EURASIP Journal on
Audio, Speech, and Music Processing
a SpringerOpen Journal

RESEARCH Open Access

The self-taught vocal interface

Bart Ons*, Jort F Gemmeke and Hugo Van hamme

Abstract

Speech technology is firmly rooted in daily life, most notably in command-and-control (C&C) applications. C&C usability downgrades quickly, however, when used by people with non-standard speech. We pursue a fully adaptive vocal user interface (VUI) which can learn both vocabulary and grammar directly from interaction examples, achieving robustness against non-standard speech by building up models from scratch. This approach raises feasibility concerns on the amount of training material required to yield an acceptable recognition accuracy. In a previous work, we proposed a VUI based on non-negative matrix factorisation (NMF) to find recurrent acoustic and semantic patterns comprising spoken commands and device-specific actions, and showed its effectiveness on unimpaired speech. In this work, we evaluate the feasibility of a self-taught VUI on a new database called DOMOTICA-3, which contains dysarthric speech with typical commands in a home automation setting. Additionally, we compare our NMF-based system with a system based on Gaussian mixtures. The evaluation favours our NMF-based approach, yielding feasible recognition accuracies for people with dysarthric speech after a few learning examples. Finally, we propose the use of a multi-layered semantic frame structure and demonstrate its effectiveness in boosting overall performance.

Keywords: Vocal user interface; Dysarthric speech; Non-negative matrix factorisation; Semantic frame description; Command and control

Introduction

Currently, modern voice control technology is available in many applications such as direct voice input (DVI) in aviation [1], information requests using Siri and speech-driven home automation. Command-and-control (C&C) appliances afford hands-free control, thus enhancing the independence of the physically incapacitated. Unfortunately, speech commands are sometimes misinterpreted when words overstep lexical boundaries and word sequences do not fit the preset grammars. Moreover, C&C appliances frequently fail to interpret dialectic or impaired speech, often encountered with physically challenged people. Consequently, people with non-standard speech are increasingly excluded from the growing market of voice-driven applications. The goal of this work is to investigate a vocal user interface (VUI) model which is able to learn words and grammars from end users, improving accessibility of C&C applications.

Over the past decade, various approaches have been proposed to improve the usability of automatic speech recognition (ASR) for speech-impaired users. For example, in [2-4], speaker-independent acoustic models were adapted to speaker-dependent and speaker-adapted models, both providing better recognition of user-specific vocalizations. Besides adaptation, dysarthric speakers also improved the recognition likelihood of their words by training the consistency of their pronunciations [5-7]; thus, users can adapt their vocalizations in order to alleviate the ASR shortcoming to cope with severe vocal variability. In [8,9], the increased phonatory variability associated with dysarthric speech was addressed by a system enabling more suitable hidden Markov model (HMM) topologies for each phoneme in the speaker's repertoire. Another example is [10], where user needs were surveyed and reflected in the design of a VUI for which an isolated word recognition system with a customizable command list and a built-in word prediction function was proposed to improve usability of typical services on mobiles and tablets. Although these approaches resulted in considerable improvements in usability, the accessibility of voice control technology still needs to widen to cater for users with non-standard or impaired speech (see [11,12]).

State-of-the-art ASR is typically based on HMM acoustic models developed with Gaussian mixture (GMM) continuous emission densities and context-dependent

*Correspondence: bart.ons@esat.kuleuven.be
Department ESAT-PSI, KU Leuven, Kasteelpark 10, 3001 Leuven, Belgium

bi- or triphone models with multiple states per model. These language-dependent models are trained on hundreds of thousands of recorded and annotated speech utterances. Some applications in voice-enabled automated home environments use ASR models together with a speaker adaptation procedure to improve ASR performance for specific users or user groups. For example, the DIRHA [13], SWEET-HOME [14] and HomeService [15] projects aim for voice-enabled assistive technology in home environments for people with a physical impairment. In the DIRHA and SWEET-HOME projects, maximum likelihood linear regression (MLLR) speaker adaptation is used starting from a speaker-independent ASR system. In the HomeService project, speaker-independent ASR models were obtained using normal or dysarthric speech followed by maximum a posteriori (MAP) speaker adaptation. These approaches require annotated language-dependent speech material in addition to annotated user-specific speech material. The advantage of the adaptation approach is that the amount of user-specific speech material composes only a fraction of the data required for building a speaker-dependent state-of-the-art speech recognizer. Speaker-dependent data usually requires an enrolment session and automated or non-automated transcriptive resources. Contrary to the adaptation approach, the basic approach here and in the (ALADIN) project (see [16] for an overview) is to build a VUI model that starts from scratch and learns from speech and demonstrations of the end user without transcription. Considering the VUI usability, the training procedure requires the ability to learn from a few examples and should be able to work with easily obtainable annotations such as content or context information. In our language-independent approach, the VUI learns to understand spoken commands by mining the speech input from the end user and the changes that are provoked on a device.

The first aim of the study is to test the feasibility of the learning procedure to construct speech patterns such as words from a few examples and content-related annotations. The speech of the user and the content information entered by the device are two sources of information that we combine by using non-negative matrix factorisation (NMF, see [17]). This procedure allows the VUI to learn co-occurring patterns from two information sources. In [18], we proposed a novel grammar induction technique based on HMM learning and semantic descriptions of commands guiding the learning process. Here, we propose multi-layered semantic structures and implement the semantic dependencies in a parse tree structure. The second aim of the study is to compare the new semantic structure with the ones employed in [18]. For this, we use two databases: one with recordings of normally speaking subjects playing a card game by voice, and another one with commands provoked in a virtual home automation setting for people diagnosed with dysarthria. The first database is referred to as PATCOR, whereas the second one is a new database called DOMOTICA-3. Besides the validation of new semantic structures, we will evaluate the NMF procedure as well by comparing our NMF-based framework with a Gaussian mixture model (GMM)-based baseline system.

The remainder of the paper is organised as follows. In the section ☒Language learning in the vocal user interface☒ we describe the learning framework, including the semantic and acoustic representations as well as the NMF learning procedure. In the section ☒Reference model☒ we describe a reference model employing GMMs instead of NMF. We proceed by describing the databases used for evaluation in the section ☒Speech material☒ Subsequently, we explain the semantic structure of spoken commands (cf. the section ☒Hierarchical knowledge representation☒) before conducting a series of experiments (cf. the section ☒Experiments☒) where we evaluate the feasibility of our approach and the effectiveness of more layered semantic structure. We present our conclusion and thoughts on a future work in the final section ☒Conclusions☒

Language learning in the vocal user interface

A schematic overview of the learning framework is depicted in Figure 1. Here, two different types of data are processed: one processing stream is depicted in the upper part and builds up a *semantic representation*, while the other one, depicted in the lower part, builds up an *acoustic representation*. In the upper processing stream, device-specific functionality is parsed into a semantic *frame description*. The conversion is guided by a hand-crafted *semantic frame structure* as indicated with the dotted arrow pointing towards the arrow leading to the block ☒frame description☒ (cf. the section ☒Semantic representation☒). The frame description is turned into a *label vector* and passed on to the *NMF* module.

In the lower part of Figure 1 (cf. the section ☒Acoustic representation☒), spectro-temporal features are extracted and transformed into mel-frequency cepstral coefficients (MFCCs, cf. the section ☒Feature extraction☒). The MFCC features are converted into a *posteriorgram*, and the horizontal dotted arrow from the right leaving from the block *Codebook/Gaussians* indicates that, for this, intermediate procedures like *codebook* training and clustering are needed (cf. the section ☒Codebook training☒). The posteriorgram is then converted into an utterance-based representation by using histograms of acoustic co-occurrence (HAC, cf. the section ☒Histogram of acoustic co-occurrence☒), after which the NMF training takes place. The depicted matrices denoted by $\mathbf{H}$ contain column-wise entries for each learning example representing loads on recurrent patterns in the data matrices $\mathbf{V}_s$ and $\mathbf{V}_a$,

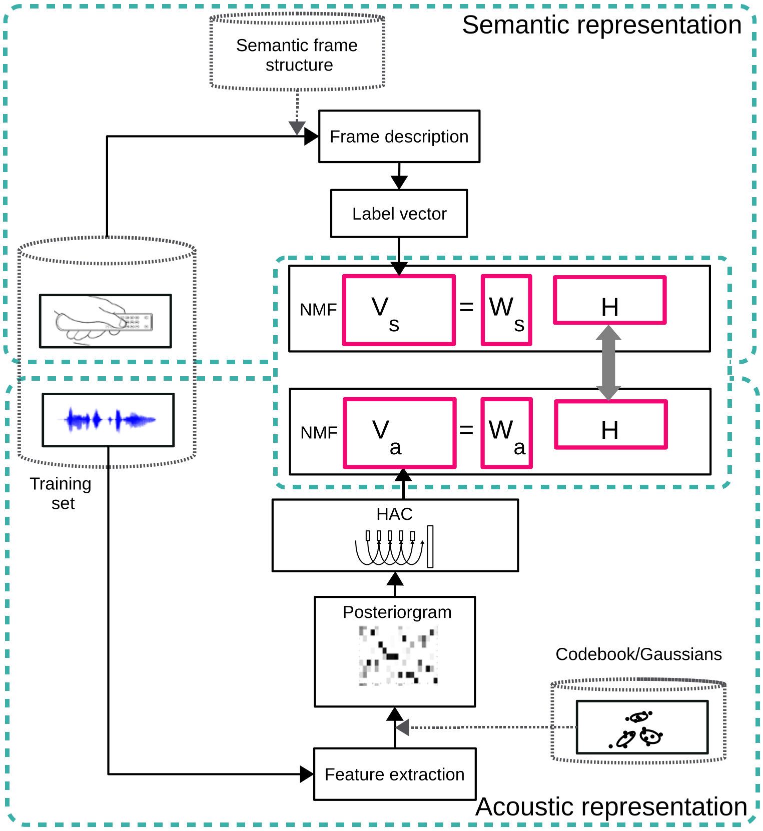

Figure 1 A schematic overview of the learning framework. Two data streams containing acoustic (lower part) and semantic (upper part) features are enhanced and processed in the direction of the arrows towards the centre where they are combined using NMF.

which are represented by the columns in the depicted matrices $\mathbf{W}_s$ and $\mathbf{W}_a$, with the subindex connoting the semantic or the acoustic stream, respectively. The large bi-directed arrow between the two matrices $\mathbf{H}$ indicates that a common matrix is sought for $\mathbf{H}$, and thus common loads on recurrent patterns which are co-occurring between the two data streams. The finding of recurrent patterns, co-occurring between the two data streams, lies at the heart of the learning procedure (cf. the section “Non-negative matrix factorisation”), where idiosyncratic expressions are parsed and linked to operations on a device. The steps and algorithms are explained with more detail in the following sections.

Semantic representation

A *semantic frame* [19] is a data structure that represents the semantic concepts in a spoken utterance which users are likely to refer when they control a device by voice. Each semantic frame is composed of slots, which in turn contain slots or values. Different commands with a similar structure are represented by the same semantic frame structure but use different slot values. For example, the correspondence between commands like “Switch off the kitchen light” and “Switch on the bathroom light” could consist of a switching *action* on the *object*, here “light”, at a particular*location*. A semantic frame with three slots, labelled by $<$ `action` $>$, $<$ `object` $>$ and $<$ `location` $>$ is a possible generic structure for parsing such commands. The values in the slots relate to the concepts describing the intended setting. Each slot allows the selection of one value from a predefined list, such as <`on, off, ...`>, <`lights, ...`> and <`kitchen, bathroom, ...`> in this example. The values also relate to the functionality of the devices, and this can be understood as a place holder with the potential capacity to hold a spoken word or phrase referring to a relevant concept. For instance, the semantic frame structure in the example above also covers commands like “Turn on the light in the kitchen”, where the spoken phrase “Turn on” is related to the value $<$ `on` $>$. The challenge is in learning to distinguish the semantic frame and filling in the correct values in the relevant slots, allowing the user to choose his own words and using his own pronunciations.

The semantic frame description of the *n*th utterance is converted into a binary *label vector*, denoted by the row vector $\mathbf{v}_{s,n}$, indicating the presence or absence of slot values collected in all frames and slots. It is a fixed-length vector with L entries equal to the total number of slot values. Note that multiple slot values are likely to be active in a single utterance and that their presence is highly correlated since the same slot values are usually active in different repetitions of the same command. Sorting all active label entries is a multi-label classification problem as multiple labels are decoded at the same time. For the collection of N utterances in the training set, the semantic representation is composed as $\mathbf{V}_s = \left[\mathbf{v}_{s,1}^T \mathbf{v}_{s,2}^T \ldots \mathbf{v}_{s,N}^T\right]$. A second utterance-based representation is built from the acoustic features as explained in the following section.

Acoustic representation

Feature extraction

The first steps in the feature extraction method are pre-emphasis and windowing followed by the fast Fourier transform. The obtained physical frequencies are rescaled to mel-frequencies which are believed to emulate the frequency scale of the human ear [20], which is approximately a linear frequency spacing below 1 kHz and a logarithmic spacing above 1 kHz. Mel-spectral magnitudes are logarithmically scaled as well and transformed into cepstral coefficients (MFCC features) by using the (inverse) discrete cosine transform. Other standard procedures in the preprocessing phase consists of voice activation detection to remove silence frames, and utterance-based mean and variance normalisation.

Codebook training

The acoustic frames are partitioned into clusters by using a codebook training procedure adopted from [17]. The procedure starts with one cluster and iteratively splits the cluster with the lowest frame sample density into subclusters. The clusters and the frames are repartitioned at each split iteration using k-means clustering. The Euclidean distance between frames was used as distance measure. The procedure continues until the requested number of K clusters is obtained. The codebook training procedure is followed by the estimation of a full-covariance Gaussian for each cluster. The set of clusters is denoted by Φ and $j = 1 \ldots K$ with K the cardinality of Φ. It is evidenced in [21] that speaker-dependent codebook training on smaller training sets is more effective than codebook training using larger training sets with speech pooled from different speakers. Therefore, we opted to use speaker-dependent codebooks for each speaker in this study.

Posteriorgram

A posteriorgram $\mathbf{P}_{t_i,\theta_j}$ is a two-dimensional data structure ($K \quad Q$) containing the posterior probabilities that the observation in the frame at time t_i, with $i = 1 \ldots Q$ and Q the number of frames, is drawn from the cluster θ_j under the assumption of a Gaussian distributed cluster membership. The posteriorgram provides a soft localization of the frame with respect to all the cluster locations in the feature space, and it contains positive values for which only a few are substantially different from zero.

Histogram of acoustic co-occurrence

The posteriorgram of an utterance has a variable length depending on the number of frames in the utterance while a fixed-length vector is required to compose a data matrix that is suitable for NMF. The aim of HAC [22] is twofold. HAC representations allow building fixed-length vectors for each utterance by accumulating the probability of observing the clusters (θ_a, θ_b) over two frames, shifted τ frames away from each other. The number of clusters is a constant; therefore, all possible $K \quad K$ co-occurring combinations for (θ_a, θ_b) are constant too. Secondly, the HAC of a posteriorgram is robust against small temporal variations because the HAC features consist of soft counts of co-occurring frames within small time delays (up to 20 frames in this study). Representations with an absolute time reference like posteriorgrams would be more prone to time-dependent variation, urging the use of time warping algorithms to compute the alignment between two time series. For the nth utterance spanning Q frames, the co-occurrence soft count over a time delay τ for the cluster pair (θ_a, θ_b) in $\Phi \quad \Phi$ is defined as follows (see [23]):

$$\left[\mathbf{v}_n^{\tau}\right]_{(\theta_a,\theta_b)} = \sum_{t_i=0}^{q \quad \tau} \mathbf{P}_{t_i,\theta_a} \mathbf{P}_{t_i+\tau,\theta_b} \tag{1}$$

and $\forall t_i, i = 1 \ldots Q, \sum_{\theta \in \Phi} \mathbf{P}_{t_i,\theta} = 1$.

The HAC is an accumulation of all the Gaussian co-occurrence probabilities denoted by the row vector $\mathbf{v}_n^{\tau}$. The information captured by the HAC depends largely on the chosen delay by which two frames are separated from each other in time. Therefore, multiple time aspects are incorporated by stacking HACs with shorter and longer delays to reach both within and across words and word boundaries. As a result, a large fixed-length column vector is built, denoted by the column vector $\mathbf{v}_{a,n} = \left[\mathbf{v}_n^{\tau_1} \mathbf{v}_n^{\tau_2} \ldots \mathbf{v}_n^{\tau_C}\right]^T$, where C represents the number of HACs. Analogous to the semantic denotation $\mathbf{V}_s$, the acoustic representation is composed as $\mathbf{V}_a = \left[\mathbf{v}_{a,1} \mathbf{v}_{a,2} \ldots \mathbf{v}_{a,N}\right]$ for the collection of N utterances in the training set.

Non-negative matrix factorisation

NMF decomposes a data matrix into the product of two low-rank matrices: one factor $\mathbf{W}$ represents latent structure, which are recurring patterns in the columns of $\mathbf{V}$, and the second factor $\mathbf{H}$ indicates which columns in $\mathbf{W}$ (patterns) are combined to approximate the columns in $\mathbf{V}$. In simultaneous NMF [24], data from different modalities are factorised simultaneously, leading to recurrent patterns in the columns of $\mathbf{W}$ consisting of pattern combinations over two or more sources that coincide with each other. Many names have been used for the same multimodal factorisation algorithm depending on the kind of source material, like for instance joint NMF in [25], or when one stream consists of supervision data, it has been referred to as semi-supervised NMF [26] or weakly supervised NMF [22].

Here, we jointly factorise the semantic and the acoustic representation in order to find the acoustic patterns that co-occur with the active label entries. The joint factorisation of $\mathbf{V}_s$ (cf. the section "Semantic representation") and $\mathbf{V}_a$ (cf. the section "Acoustic representation") is expressed as follows:

$$\begin{bmatrix} \mathbf{V}_s \\ \mathbf{V}_a \end{bmatrix} \approx \begin{bmatrix} \mathbf{W}_s \\ \mathbf{W}_a \end{bmatrix} \mathbf{H} \tag{2}$$

where $\mathbf{W} = [\mathbf{W}_s \mathbf{W}_a]^T$ and $\mathbf{H}$ are two matrices of lower rank. The co-occurring semantic and acoustic patterns are found in $\mathbf{W}_s$ and $\mathbf{W}_a$, respectively. The nth column in $\mathbf{H}$ describes which co-occurring patterns are active in the nth utterance. The inner dimension in the right half of Equation 2 determines the number of co-occurring patterns in which the dataset is decomposed. It is usually a low number in a small vocabulary task since it reflects the number of slot values L in the VUI. However, by increasing the inner dimension with a number D, columns are introduced in $\mathbf{W}$ to represent patterns for filler words, i.e. recurrent acoustic patterns such as "please" or "the" that are usually left out in the semantic representation.

The latent patterns are found by minimising the difference between both sides of Equation 2. Since $\mathbf{V}_s$ and $\mathbf{V}_a$ consist of (soft) count data, the Kullback-Leibler divergence [27] is preferred as loss function and is expressed as follows:

$$\left(\mathbf{H}^*, \mathbf{W}_a^*, \mathbf{W}_s^*\right) = \underset{(\mathbf{H}, \mathbf{W}_a, \mathbf{W}_s)}{\arg\min}\, D_{KL}\left(\begin{bmatrix} \mathbf{V}_s \\ \mathbf{V}_a \end{bmatrix} || \begin{bmatrix} \mathbf{W}_s \\ \mathbf{W}_a \end{bmatrix} \mathbf{H}\right) \tag{3}$$

Iterative update rules for minimising a distance measure between the left- and the right-hand side can be found in [27]. It has been demonstrated that convergence is guaranteed towards a local optimum. Note that the loss function in Equation 3 can be seen as a regularisation in which acoustic patterns are preferred that correspond to the occurrences of slot values. Writing the loss function in Equation 3 as a regularised loss function results in

$$\begin{aligned}\left(\mathbf{H}^*, \mathbf{W}_a^*, \mathbf{W}_s^*\right) = \underset{(\mathbf{H}, \mathbf{W}_a, \mathbf{W}_s)}{\arg\min}\, & [D_{KL}\left(\mathbf{V}_a || \mathbf{W}_a \mathbf{H}_a\right)) \\ & + \lambda D_{KL}\left(\mathbf{V}_s || \mathbf{W}_s \mathbf{H}_s\right)]\end{aligned} \tag{4}$$

with $\lambda = 1$ and $\mathbf{H}_a = \mathbf{H}_s$ for equivalence with Equation 3. If $\mathbf{H}_a$ and $\mathbf{H}_s$ are allowed to be tied loosely, then an additional regularisation term should be added to minimise the difference between $\mathbf{H}_a$ and $\mathbf{H}_s$, which was pursued in [25].

Recognition

The aim of the VUI is to find the frame description for a spoken utterance. A schematic overview of the recognition phase is depicted in Figure 2. Speech processing of a command proceeds from the spectro-temporal representation in the lower part of Figure 2 to the HAC representation in the centre, after which NMF takes place in order to obtain the load matrix $\mathbf{H}_t$ using the learned patterns in $\mathbf{W}_a$ that were co-occurring with the semantic patterns in $\mathbf{W}_s$ in the training phase. $\mathbf{H}_t$ is then transferred to the upper part of Figure 2. The slot value activations $\mathbf{A}$ are found by using $\mathbf{H}_t$ and using $\mathbf{W}_s$ obtained in the learning phase. Finally, the arrow leaving from the box "Semantic frame structure" indicates that the semantic structure is superimposed on slot value activations as a decision process where groups of slot values are compared and related to each other (cf. the section "Decision process"). Knowing the correct frame description of the spoken command allows for the proper execution of the command.

Activation

We denote the data matrix and the load matrix in the test phase by $\mathbf{V}_t$ and $\mathbf{H}_t$. The data matrix $\mathbf{V}_t$ contains the processed speech signal, and $\mathbf{H}_t$ is found by minimising the Kullback-Leibler divergence between $\mathbf{V}_t$ and the matrix product of the acquired $\mathbf{W}_a^*$ and the unknown $\mathbf{H}_t$.

$$\mathbf{H}_t^* = \arg\min_{\mathbf{H}_t} D_{KL}\left(\mathbf{V}_t || \mathbf{W}_a^* \mathbf{H}_t\right) \tag{5}$$

Contrary to Equation 3, an optimal solution for $\mathbf{H}_t^*$, given $\mathbf{W}_a^*$, is guaranteed since the loss function in Equation 5 expresses a convex problem. The obtained matrix $\mathbf{H}_t^*$ and the acquired matrix $\mathbf{W}_s^*$ are used to provide the slot value activations in $\mathbf{A}$,

$$\mathbf{A} = \mathbf{W}_s^* \mathbf{H}_t^* \tag{6}$$

Note that the last step in Equation 6 allows the freedom to obtain slot value activation from different latent factors in $\mathbf{W}^* = \left[\mathbf{W}_s^* \mathbf{W}_a^*\right]^T$. A slot value activation can depend on one latent factor or a combination of latent factors in $\mathbf{W}^*$.

Decision process

The decision whether a particular slot value applies or not depends on the ensemble of activations signalling the presence of the related frame, slots and values. The interdependent relation between frames, slots and values is vital information that we use on top of the activations obtained from Equation 6. The whole semantic frame structure is taken into account by implementing a parse

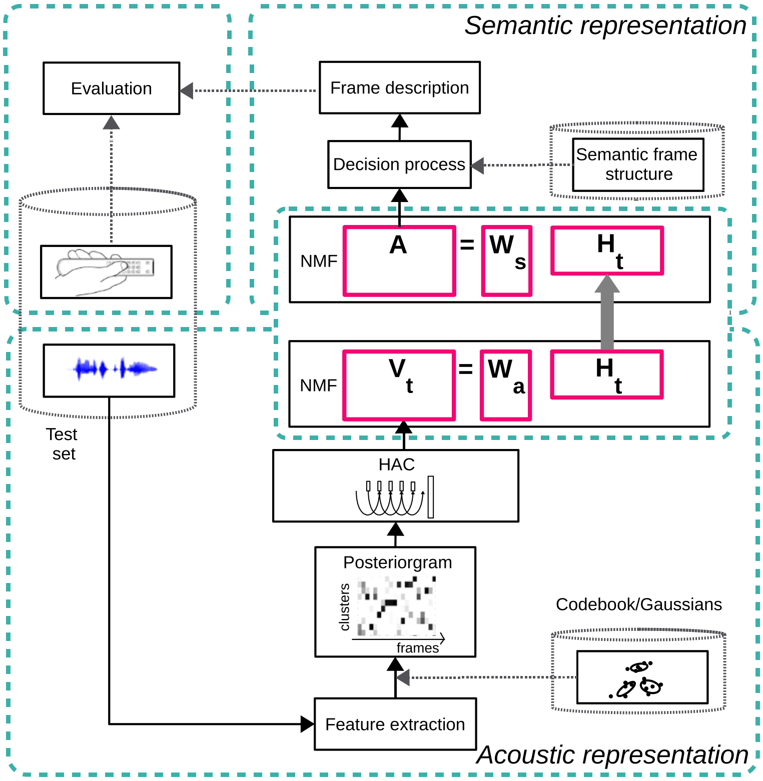

Figure 2 A schematic overview of decoding. Only acoustic data is available, and the processing proceeds from the bottom to the upper part where a decision process takes place to validate the interdependent activations of different words.

tree for each frame and a few activation spreading rules. In the parse tree (see Figure 3), the frames are considered root nodes, slots compose branch nodes, and slot values compose leaf nodes. Activations in leaf nodes correspond with the activations in the columns of **A**, denoted by the vector **a** and representing the NMF activations of one utterance. For each entry in **a**, there is one leaf node[a]. The activation spreading rules enable values in **a** to propagate to slot and frame levels. These activation spreading rules bear on child-parent relations to which we refer as ‘exclusive’ and ‘selective’ relations. In an *exclusive* relation, only a predefined number of entries, denoted by the constant U, occur at the same time. For instance, the light can be switched on or off, but not both at the same time; therefore, a command can only be assigned one true value ($U = 1$) in the list `<on, off>`. Generally, if we denote the activation of a parent node by a_p and the activations of its child node by a_{ci}, with $i = 1, \ldots, z$ and $a_{c1} > a_{c2} > \ldots > a_{cz}$, then the following activation spreading rule applies for an exclusive relation: $a_p = \text{median}\left(\{a_{ci} | a_{ci} \geq a_{cU}\}\right)$. A selective relation differs from an exclusive one in the precognition of the number of valid child nodes. If the number of valid child nodes is unknown, then activations are compared against a threshold. The following general activation spreading rule applies for a selective relation: $a_p = \text{median}\left(\{a_{ci} | a_{ci} \geq a_0\}\right)$ and a_0 is a threshold determined by the pth percentile of all activations in **a**. Multiple frame descriptions are in competition, and the frame description with the highest activation in the root node is selected. The U highest activated slots and values for exclusive relations and the slots and values scoring higher than a_0 for selective relations are included in the predicted frame.

A toy example is depicted in Figure 3 where the first nine entries of **a** are set to $[0.4, 0, 0.1, 0, 0, 0, 0.5, 0, 0.1]^T$ corresponding to the frame description listed in the upper half of Table 1. Exclusive relations are depicted as solid arrows, and selective relations are depicted as dashed arrows. The first slot value has activation 0.4 and corresponds to the empty slot value of the frame ‘increase heating’. The activation is propagated to the slot level and from there to the frame level. Exclusive relations are presumed for the slots `<action>` and `<object>` and their respective slot values. A preset value of $U = 1$ for both slots yields the propagation of the highest activation, $a_p = 0.1$ and $a_p = 0.5$, to the `<action>` and `<object>` slots, respectively. The relation between the ‘open_close’ frame and its slots is evaluated as selective, and by assuming a preset threshold beneath 0.1, their median, $a_p = 0.3$, is propagated to the frame level ‘open_close’. Generally, the

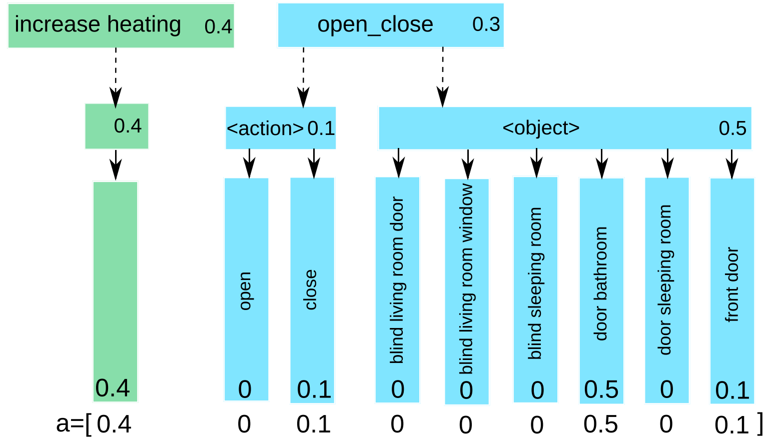

Figure 3 Parse tree and propagation of activation. A parse tree of the first two frame descriptions listed in Table 1 and the propagation of activation depending on exclusive and selective relations. See text for more explanations.

median is an unbiased measure for propagating activations when the number of slots and values differs between different frames. In this hypothetical example, the frame “increase heating” and “open_close” have activation 0.4 and 0.3, respectively; thus, the frame “increase heating” and its selected slots and slot values are the predicted outcome of the spoken utterance yielding the activations **a** in this toy example.

Reference model

We compared NMF learning with GMMs. It is hard to preset the number of GMM components especially when datasets are small and have varying sizes among speakers (see the subsequent section “Speech material”). Therefore, we investigated four GMMs having 10, 20, 40 or 80 components fixed for each speaker, respectively, with a diagonal covariance structure instead of a full one in order to limit the number of free parameters. The GMMs were embedded in the architecture of our framework in a similar way as the NMF learning module. At the front end, feature extraction was identical up to and including the posteriorgram step (see Figure 1), after which a scaling step was introduced using the logit function, $\log p/(1 - p)$, to map probabilities to the real line $\mathbb{R}$. Subsequently, utterances in the data with a common semantic entry, i.e. utterances with “1” at a particular position in the label vector $\mathbf{v}_s$, were pooled together to compose the training set for GMM estimation of each respective slot value. Thus, for each label entry in $\mathbf{v}_s$, there is one GMM predicting the presence of the respective slot value in the decoding phase. Similar to the NMF activations (see the section “Activation”), the posterior probabilities are committed to the same decision process (cf. the section “Decision process”). It should be noted that GMMs do not capture temporal dependencies while HACs do. A GMM can be conceived as a HMM with one state per slot value. By using a HMM with multiple states and tuning transition probabilities, temporal relations among the acoustic features can be captured. However, it is evidenced in [28] that GMMs outperform HMMs in feasibility for small training sets.

Table 1 Semantic frame descriptions for DOMOTICA-3 - compositional

Frame (exclusive)	Slot (selective)	Value (exclusive)
Increase heating	-	-
Open_close	< action >	open,close
	< object >	1-6
Ranged	< *range* >	1,2,3
	< object >	1,2
on_off	< *action* >	on,off
	< object >	1-6

The numbers 1-6 refer to objects such as a kitchen lamp or a bathroom door.

Speech material

Similar to [18], two datasets are employed. The first dataset is PATCOR containing recordings of ten speakers playing a solitaire card game by voice. The second dataset is a recent recorded dataset called DOMOTICA-3 dubbing its precursor DOMOTICA-2 employed in [18]. The utterances consist of commands controlling a home automation system by voice.

PATCOR

The database PATCOR contains recordings of subjects playing the card game “patience” on a computer, using only spoken commands. The database contains ten speakers with more than 2,000 commands. The data was collected from unimpaired subjects with non-pathological speech, speaking Belgian Dutch. As depicted in Table 2, six participants were females and the age ranged between 22 and 45 years old for almost all speakers except for speaker 9 who was 73 years old. All players played at least four games leading to 254 recorded utterances on average, except for the speaker in the second entry who played only two games.

In order to provoke commands in a natural human-machine-like interaction, a wizard-of-Oz setup was employed for five players as indicated in column 4 of Table 2. In a wizard-of-Oz setup, a subject is deceived to believe that the machine is able to commit responsive behaviour, while in reality, the administrator is taking care of the responsive actions of the machine. The five other players in PATCOR were committed to the same procedure; however, they were told that the administrator took care of all the actions.

The users were free to choose their own words and grammars allowing different expressions for the same card move. A typical utterance in PATCOR is "Put the four of clubs on the five of hearts". The standard frame structure of the utterances used in [18] is demonstrated in Table 3.

Table 2 Participants in PATCOR

PID	Gender	Age (years)	Wizard-of-Oz	Number of games	Number of utterances
1	♀	33	Yes	6	274
-	♀	41	Yes	2	169
2	♂	45	Yes	4	260
3	♂	42	Yes	5	278
4	♀	23	No	4	222
5	♀	26	No	4	248
6	♂	24	No	4	223
7	♂	26	No	4	240
8	♀	73	No	5	235
9	♀	22	Yes	5	262

Domotica-3

The DOMOTICA-3 database contains Dutch dysarthric speech commands related to home automation. A typical DOMOTICA-3 utterance is "turn on the kitchen light". The dataset contains recordings of the speakers that participated in the collection of the DOMOTICA-2 dataset in [18].

In short, a two-phase data collection method was used in DOMOTICA-2. In the first phase, nine users were asked to command 27 distinct actions (see Table 4) in a 3D home environment on a computer, guided by a visualised and narrative scenario such as "you enter the kitchen, but it is dark...", in order to provoke an action, but to ensure an unbiased choice of words and grammar. Consequently, each user produced a list of natural induced commands; thus, nine different lists of commands controlling the same actions were created. Some participants missed out a few actions during the guidance of the narrative scenario, but never more than two. The lists were read repeatedly by multiple speakers in the DOMOTICA-2 data collection.

Table 3 Semantic frame descriptions for PATCOR-compositional

Frame (exclusive)	Slot (selective)	Value (exclusive)
Dealcard	-	-
Movecard	`<from_suit>`	c,d,h,s
	`<from_value>`	1-13
	`<from_foundation>`	1-4
	`<from_column>`	1-7
	`<from_hand>`	-
	`<target_suit>`	c,d,h,s
	`<target_value>`	1-13
	`<target_foundation>`	1-4
	`<target_column>`	1-7

Here, the letters c,d,h and s represent the suits clubs, diamonds, hearts and spades, respectively.

A selection of 27 actions from the DOMOTICA-2 collection were used in the new recordings of the DOMOTICA-3 database. A recording session lasted more or less a half hour in which the speaker read repeatedly the commands from one of the nine lists (see the fifth column in Table 5). To keep correspondence with the previous and future work, we refer to these speakers by unique IDs. For all adult speakers, speech intelligibility scores were obtained by analysing the recorded speech using the automated procedure in [29]. While a score above 85 is considered as normal speech intelligibility, a score equal to or below 70 is considered as severely impaired. Speaker characteristics are listed in Table 5. Speakers 31 and 37 were children and did not conduct an intelligibility test. Additionally, speakers 43, 44, 46, 47 and 48 were diagnosed as multiple sclerosis patients, and some of them demonstrated adequate speech intelligibility. They were recruited because the digressive nature in time of their speech ability would allow for speech-degenerating data collection in the future. Most speakers were able to generate six or more repetitions of the command lists, except for speakers 31 and 47 who were able to produce one and two repetitions, respectively. A few speakers received a reduced list with ten commands with at least ten repetitions. A larger number of repetitions allow us to investigate whether learning improvements proceed even further by adding more learning examples, or whether it levels off at a particulate stage. The number of utterances

Table 4 Synoptic description of all actions in DOMOTICA-3, partitioned in columns according to frame type

Increase_heating	Open_close	Ranged	On_off
Increase_heating	close_blind_living_room_door	dimstate1_floor_lamp	off_light_living_room
	close_blind_living_room_window	dimstate2_floor_lamp	off_light_sleeping_room
	close_blind_sleeping_room	dimstate3_floor_lamp	off_lights
	close_door_bathroom	level1_head_bed	on_light_bathroom
	close_door_sleeping_room	level2_head_bed	on_light_kitchen
	close_front_door	level3_head_bed	on_light_living_room
	open_blind_living_room_door		on_light_sleeping_room
	open_blind_living_room_window		on_reading_light
	open_blind_sleeping_room		
	open_door_bathroom		
	open_door_sleeping_room		
	open_front_door		

is indicated in column 6 of Table 5. The frame description used in [18] is displayed in Table 1.

The database contains speech recorded in realistic environments with two microphones. One microphone was a head-worn set C520, and the other one was a RØDE M2 live condenser microphone, which was located in front of the speaker on top of a table at about 50 to 100 cm. The recordings were held in a room selected in the respective health care centre of the patient which ranged from quiet to some background speech. The recordings were carried out with a sampling rate of 48 kHz and a resolution of 24 bit for each channel, after which it was downgraded to a sampling rate of 16 kHz and stored as such in the corpus. The recordings of speakers 33 and 40 barely reached voice activation levels because the directed microphone of the headset was too far out of reach; however, the recordings on the second channel did succeed.

Hierarchical knowledge representation

An optimal structure depends on different factors like the number of decision steps in the recognition process (cf. the section “Decision process”) and the complexity for each decision depending on the number of alternatives. These factors are not independent from each other. For instance,

Table 5 Participants in DOMOTICA-3

PID	Gender	Age (years)	Profile	Spoken list number	Number of utterances	Intelligibility score
17	♀	25	Spastic quadriparesis	6	347	88.6
28	♀	42	Severe nasal dysarthria	6	204	73.1
29	♂	44	Spastic quadriparesis	7	174	73.6
30	♂	33	Ataxic dysarthria	5	198	69
31	♂	11		8	225	
32	♀	43	Mild dysarthria and hyperkinetic speech	4	41	65.6
33	♂	33	Ataxic dysarthria, short phonation	3	113	66.2
34	♂	61	Multiple sclerosis	6	331	76.2
35	♀	25	Spastic quadriparesis	4	268	72.3
37	♂	10		8	156	
40	♂	55	Myotonic-flaccid dysarthria	1	184	85.5
41	♀	39	Dysarthria	2	144	64.2
43	♀		Multiple sclerosis	1	133	89.4
44	♂		Multiple sclerosis	9	164	89.2
46	♀	50	Multiple sclerosis	1	97	74.9
47	♂		Multiple sclerosis	7	64	73.4
48	♂		Multiple sclerosis	5	169	85.8

an ordered tree with more levels will induce more decisions, but with lower complexity. Semantic structures allowing for overall simple decisions improve slot value perplexity. In this study, we explore the influence of the semantic frame composition on the recognition performance by considering two different frame structures employing different hierarchical levels and decision rules.

We will investigate two approaches for the validation of frame structure on the database PATCOR; one is the compositional standard shown in Table 3 and employed in [18], and the second one is a hierarchical semantic frame structure with one additional level shown in Table 6. The decision rules for each layer are listed in the column headings. In the standard description, commands are decomposed into parts such as suits, values and columns. In the *compositional* structure, a selective rule is used to compare the activations of alternative slots against a threshold. Known information is left unexplored like for example the impossible co-occurrence of a card moved from the hand and from the foundation. A second structure is called *hierarchical* referring to more levels in which slots contain slots or values. Such a structure alleviates the decision step as the number of alternatives is limited in each layer with no more than two alternatives in selective decisions.

For the DOMOTICA-3 dataset, we employ even more distinctive structures on the semantic representation. The first structure entails the mapping of entire spoken commands to frames without slots or values, leading to a scenario where the machine learning problem reduces to a multi-class paradigm, that is one class for each possible command. Clearly, such a mapping is unattainable for sets with complex commands as in PATCOR, but for a small set of commands, modelling entire utterances is a viable option. Note that such a structure is less robust to word order variation, and alternative expressions of the same command as utterances are learned in their entirety. We compare a semantic frame structure with commands modelled in their entirety with a compositional approach which parses commands into parts such as objects and actions [18]. This semantic frame structure is shown in Table 1. The values 1 to 6 refer to objects or devices such as kitchen lamp or bathroom door. Once again, we expect improved performance for the multi-layered frame structure since selective rules are used for layers holding only two slots, while multiple alternatives are gathered in levels with exclusive rules.

Table 6 Semantic frame descriptions for PATCOR - hierarchical

Frame (exclusive)	Slot (selective)	Slot (exclusive)	Slot (selective)	Value (exclusive)
Dealcard	-	-	-	-
Movecard	< from >	< card >	< suit >	c,d,h,s
			< value >	1-13
		< foundation >	-	1-4
		< column >	-	1-7
		< hand >	-	-
	< target >	< card >	< suit >	c,d,h,s
			< value >	1-13
		< foundation >	-	1-4
		< column >	-	1-7

Here, the letters c,d,h and s represent the suits clubs, diamonds, hearts and spades, respectively.

Experiments

The goal of the experiments is twofold: first, we test the feasibility of our VUI by evaluating the performance of the framework using the F-score on slot value recognition as defined in [18]; furthermore, we investigate the added value of using a more layered semantic frame structure on two datasets: PATCOR containing commands having a complex grammar and DOMOTICA-3 containing realistic recordings of commands from speech-impaired speakers in the setting of a virtual home automation system.

An important feasibility issue is the speed of learning, which we evaluate by tracking the gain in slot value recognition for incrementally increasing training sets. This procedure allows us to plot a learning curve, that is, the curve representing the average slot value recognition score in function of the average number of learning examples. The rate of learning is usually sharpest in the beginning and gradually evens out against an asymptotic level. We are especially interested in the initial and final phase of the learning curve; on the one hand, the speed of learning should be high so users gain interest in keeping on using the VUI, thus keeping on training the system; on the other hand, the learning curve should not level off to low, that is, the VUI should not get stuck in suboptimal functionality in the long run. Clearly, the speed of learning and the asymptotic performance are important attributes of a useful learning procedure.

Setup

Evaluation procedure

The data was partitioned in blocks containing approximately an equal number of slot values using an algorithm outlined in [18]. This algorithm minimises the Jensen-Shannon divergence between the slot value distributions over all blocks. Likewise [18], block creation was followed by the composition of a Latin square from which the first five rows were submitted to a fivefold cross-validation experiment. In each fold or row of the Latin square, the first x blocks were used as train set while the remaining Z x blocks were used as test set with Z as the total number of blocks. While the train sets increased incrementally with one block, $x = 1, \ldots, Z \quad 1$, the test sets decreased decrementally with one block. The incrementally increasing training sets allowed us to evaluate the learning performance at different time stamps in the learning process of the VUI. The slot values that appeared in each block at least once were used for scoring in the test sets. Note that the real performance of the vocal interface also depends on the interface⊠s ability to distinguish between commands and other utterances spoken in a domestic environment. Here, we focus on the rate of learning assuming a perfect classification of commands directed to the system against utterances that were not.

For the evaluation of the framework, we excluded the speaker without PID number in Table 2 in PATCOR and speakers 32 and 47 in DOMOTICA-3, due to data insufficiency for block creation. In addition, we created two groups for the DOMOTICA-3 corpus in order to evaluate the feasibility of the framework. Speakers $29, 30, 33, 41$ and 46 have an intelligibility score below 75 and uttered less than 200 commands. We refer to this group as *severe dysarthria*. Note that an intelligibility score higher than 85 is not considered pathologic. Speakers $17, 28, 31, 34$ and 35 were joined in another group because they uttered more than 200 commands allowing us to track the performance of the system in the long run.

Parameters

We used pre-emphasis ($\alpha = 0.97$, sampling rate at 16 kHz) and Hamming windowing with 30-ms frames in addition to a frame shift of 10 ms. Fourteen cepstral dimensions were retained, and the first and second order differences were appended, leading to 42 feature dimensions. Silence frames were removed before the codebook training started, aiming for $K = 100$ clusters from which posteriorgrams were obtained with 100 entries. The main portion of the probability mass in a frame seems to originate from only a few clusters; therefore, we retained only the three highest probabilities in each frame in order to gain computational efficiency by using sparse matrices HAC features. We stacked $C = 4$ HACs with delays $\tau = 2, 5, 9$ and 20 resulting in $4 \quad 100^2 = 40,000$ entries for each utterance-based acoustic representation $\mathbf{v}_a$.

$\mathbf{H}_{\text{init}}$ and $\mathbf{W}_{\text{init}}$ denote the initialisation of $\mathbf{H}$ and $\mathbf{W}$, respectively,

$$\mathbf{H}_{\text{init}} = \begin{bmatrix} \mathbf{V}_s + \lambda \mathbf{A}(R \quad N) \\ \mathbf{B}(D \quad N) + \gamma \mathbf{1}(D \quad N) \end{bmatrix} \tag{7}$$

$$\mathbf{W}_{\text{init}} = \begin{bmatrix} \mathbf{I}(R \quad R) + \lambda \mathbf{O}(R \quad R) & \mathbf{P}(R \quad D) + \theta \mathbf{1}(R \quad D) \\ \multicolumn{2}{c}{\mathbf{Q}(F \quad (D+R)))} \end{bmatrix} \tag{8}$$

with D being the largest integer smaller than $0.2 \quad R$; hence, by way of example, for $R = 40$ slot values, $D = 8$ extra columns were added to $\mathbf{W}$. This proportion was constant for all experiments. The parameters λ, γ and θ were set to $1e \quad ^4$, 0.1 and 0.2, respectively. All entries in $\mathbf{A}, \mathbf{B}, \mathbf{O}, \mathbf{P}$ and $\mathbf{Q}$ are i.i.d samples from the uniform distribution $\mathcal{U}$ with boundaries $(0, 1)$. $\mathbf{I}$ is the identity matrix and $\mathbf{1}$ is a matrix with all ones. The columns of $\mathbf{W}$ were normalised to sum to one throughout the multiplicative updates to prevent drift towards large numbers reducing the cost function.

Results and discussion

Feasibility

Results on DOMOTICA-3 are shown in Figure 4 as a function of the number of learning examples in the training set. The depicted results concern recordings on the field microphone. The F-scores for the more severe dysarthric group are depicted in the upper panel. The plotted numbers are the PIDs of the speakers (see Table 5) with circle-shaped lighter and darker gray background colours indicating the NMF-based and the 80-component GMM approach, respectively. When we compare GMM-based learning with NMF-based learning in the upper panel, we observe steeper learning curves for the NMF-based learning for the group of severely dysarthric speakers, yielding an average improvement of 23% ($t_{(159)} = 30.2, p < 0.001$). Moreover, a similar trend can be observed in the group with more training material depicted in the lower panel of Figure 4, yielding an average improvement of 20.2% ($t_{(159)} = 38.5, p < 0.001$) by using the NMF-based approach.

We used the non-parametric method in [30,31] to estimate a smooth learning curve for each speaker using the locally weighted scatterplot smoothing (LOWESS) procedure. Optimal smoothness parameters were found by cross-validating different smoothness values between 0.4 and 0.8. We plotted the learning curve for the average speaker using a full blue-coloured and a dashed green-coloured line to indicate the NMF-based and the GMM-based scores, respectively. Furthermore, we constructed 95% confidence limits by bootstrapping the LOWESS procedure, and we indicated these bounds by dotted lines. The curves provide an indication of how the average speaker is expected to perform.

When comparing F-scores, the gain by using NMF learning is higher in the beginning of the learning curve, up to 40% absolutely on average, as can be seen from the difference between the full and the dashed-line averaging curve. For the group with severe dysarthria, we observe that the NMF-based approach yields a score close to 80% on average after only one repetition. For instance, speaker 33 has an intelligibility score of 66.2 and yields an F-score of 70% after one repetition and 96% after nine repetitions.

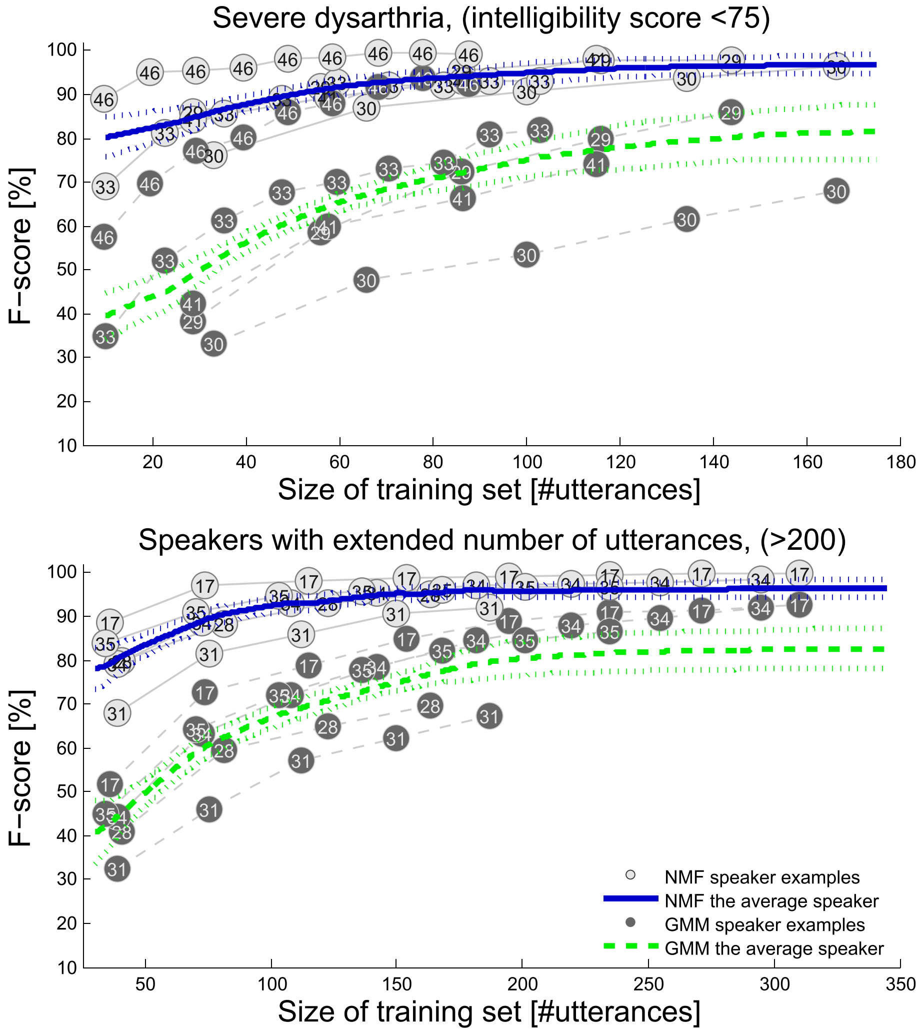

Figure 4 NMF-based learning against GMM-based learning. Upper part: for severe dysarthric speakers. Lower part: for speakers with extended training sets in the lower part. Numbered circles represent PID, and their locations indicate F-scores as a function of the number of utterances in the training sets. Furthermore, the smoothed curves are interpolations of the scattered F-scores using the LOWESS procedure, and they exemplify the performance of an average speaker.

Moreover, some speakers yield scores close to 100% after a few repetitions, like for instance speaker 17 depicted in the lower part of Figure 4 obtaining a score above 99% after four repetitions only. Note that the results using the headset recordings are similar as can be seen in Table 7. These results are very promising, especially for dysarthric speakers, as both the learning rate and the accuracy are already in a range that is usable for a vocal interface. Moreover, all learning curves that did not reach a ceiling performance at the end are still rising, indicating that with more learning examples, the accuracies will probably further improve.

Semantic structure

Here, we compare the results for NMF-based learning using two different semantic frame structures on the PATCOR database depicted in the upper panel and the DOMOTICA-3 database depicted in the lower panel of Figure 5. When comparing F-scores for the hierarchical and the compositional frame structure in Figure 5, we find a small but significant overall improvement for using a hierarchical frame structure instead of a compositional one, i.e. $t_{(179)} = 12.4, p < 0.001$, with an absolute average improvement in F-score of 3.3%. The improvements are fairly consistent among speakers despite the fact that the individual scores for the PATCOR database are wide ranged. The scores are wide ranged because speakers $3, 5, 7$ and 8 frequently used the words red and black instead of hearts, spades, clubs and diamonds While the use of colours such as red and black allows the VUI to distinguish clubs and spades from hearts and diamonds, it will not allow to learn the

Table 7 F-scores after 40 and 120 training utterances for DOMOTICA-3

				Speakers															Average
				17	**28**	**29**	**30**	**31**	**33**	**34**	**35**	**37**	**40**	**41**	**43**	**44**	**46**	**48**	
DOMOTICA-3 RØDE M2	Compositional		GMM 10	60	51	53	41	43	58	52	53	62	43	53	88	45	83	77	57.5
			GMM 80	53	45	47	34	34	65	46	49	56	37	46	87	37	81	79	53.1
			NMF	90	69	83	82	71	87	76	83	84	73	77	99	75	96	98	82.9
		$N = 120$	GMM 10	75	66	66	51	52	67	68	66	69	57	66	88	75	88	85	69.3
			GMM 80	80	65	76	56	59	82	74	73	79	69	68	97	78	96	93	76.3
			NMF	99	88	90	93	86	93	91	94	94	92	96	100	99	97	99	94.1
	Flat	$N = 40$	GMM 10	43	37	43	23	25	41	38	43	48	29	35	78	62	73	66	45.6
			GMM 80	24	19	27	18	12	45	23	31	42	16	28	84	57	75	74	38.3
			NMF	88	72	80	76	63	78	79	81	77	71	68	99	98	96	98	81.6
		$N = 120$	GMM 10	65	55	65	39	39	48	61	57	51	51	48	91	88	82	81	61.4
			GMM 80	65	50	74	49	35	67	59	61	65	53	60	97	97	85	87	66.9
			NMF	98	83	95	94	78	85	90	89	88	86	93	100	100	100	99	91.9
DOMOTICA-3 headset	Compositional	$N = 40$	GMM 10	58	45	54	42	42	40	45	53	60	35	56	87	59	86	82	56.3
			GMM 80	54	42	49	37	33	51	46	49	58	31	50	87	54	82	82	53.7
			NMF	89	80	89	80	69	79	80	85	86	60	88	99	90	96	98	84.5
		$N = 120$	GMM 10	74	56	69	48	52	43	63	69	65	46	69	87	74	90	88	66.2
			GMM 80	79	65	79	57	58	70	74	75	79	55	75	97	89	96	92	76
			NMF	98	92	97	92	86	92	94	96	95	87	98	100	99	99	100	95
	Flat	$N = 40$	GMM 10	41	29	39	24	24	25	36	41	45	25	37	82	70	73	75	44.4
			GMM 80	27	18	32	18	13	31	23	30	38	11	30	85	59	75	74	37.6
			NMF	88	79	90	78	63	66	83	87	79	52	84	98	98	98	99	82.8
		$N = 120$	GMM 10	64	48	67	45	38	32	57	55	48	38	61	80	89	80	80	58.8
			GMM 80	70	45	73	52	39	52	61	58	64	39	63	96	97	82	89	65.3
			NMF	98	88	97	94	82	92	94	92	92	74	98	100	100	100	100	93.4

The F-scores are interpolated using the LOWESS procedure.

difference between the two black or the two red card suits. Since 40% to 50% of the words in the move commands consisted of words referring to the card suits, a drop in overall F-score is observed because the incorrectly recognised card suits are counted as false positives despite the fact that the user did not provide this information in the VUI training. As can be seen in the upper part of Figure 5, there is a considerable gap between the learning curves of speakers $3, 5, 7$ and 8 using the words “red” and “black” and speakers 2,4,6 and 9 who all preferred the consistent use of the words “clubs”, “spades”, “hearts”, and “diamonds”. Another reason for the wide-ranged performances is that some users tend to use a lot of synonyms, which we did not anticipate in the NMF-based approach here. More results on the PATCOR database, including results on GMMs, are listed in Table 8. Note that GMMs with more

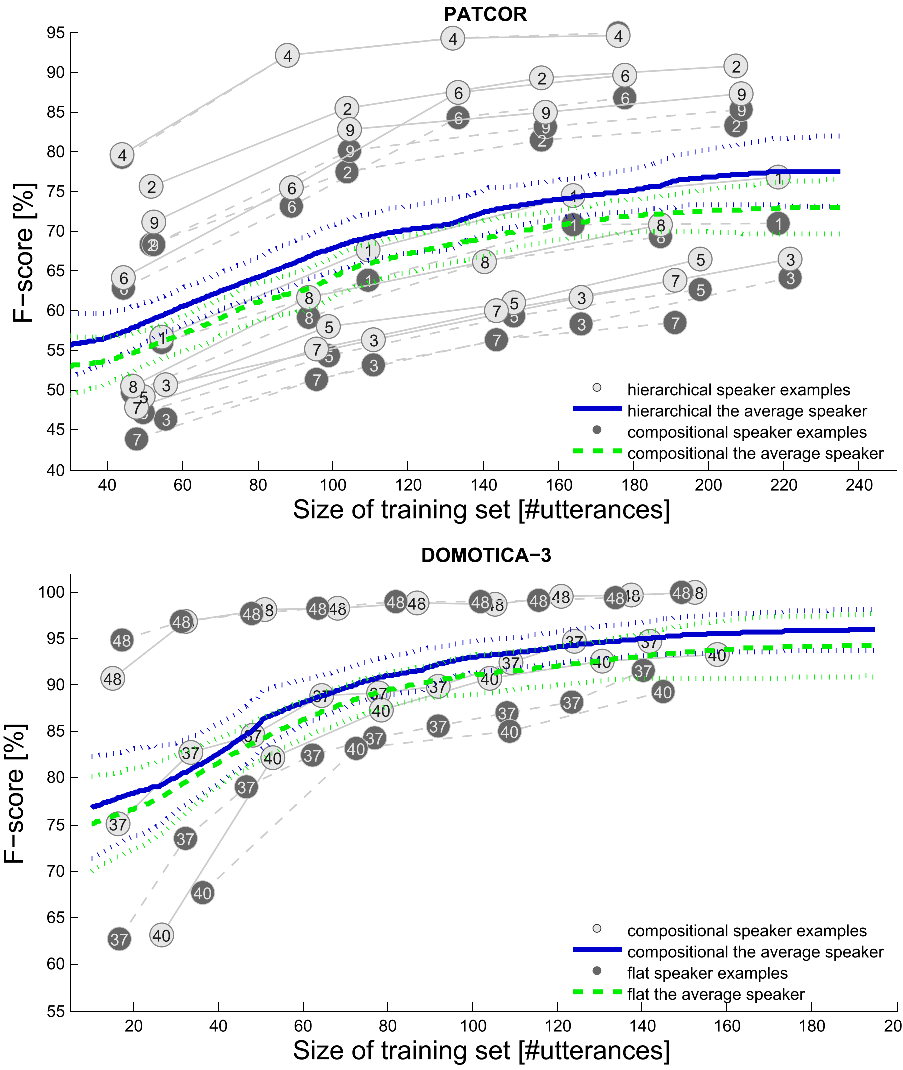

Figure 5 Hierarchical, compositional and flat structures. Hierarchical against compositional frame structure for PATCOR in the upper part, and the compositional against the flat structure for DOMOTICA-3 in the lower part. Numbered circles represent speaker ID, and their locations indicate F-scores as a function of the number of utterances in the training sets. Furthermore, the smoothed curves are interpolations of the scattered F-scores using the LOWESS procedure, and they exemplify the performance of an average speaker.

compounds perform better if there is enough data to adequately fit all free parameters as can be seen in Tables 7 and 8 when comparing GMM scores for small datasets ($N = 40$) against large datasets ($N = 120$ or $N = 175$). As can be seen in the tables, the GMM with 80 components demonstrates better performance than the GMM with 40 components for $N = 175$, but not for $N = 40$. These tables include GMM scores for 10 and 80 components. The GMM results for GMMs with 20 and 40 components are not reported here because these scores are similar to the 80-component GMM scores.

The corresponding results for the DOMOTICA-3 database are depicted in the lower panel of Figure 5, displaying a positive, though, non-significant statistical tendency in favour of the compositional frame structure, i.e. the more profound structure compared to the flat one. The average speaker plot represents scores of all 15 speakers in the database, though the varying range of results is exemplified by three speakers only for reasons of visibility. A considerable number of speakers yield high F-scores in the beginning while other speakers yield lower F-scores in the beginning, but a steeper rise towards the end, as demonstrated by speakers 48 and 37, respectively. The non-significant statistical tendency is probably caused by the ceiling effect, in which a considerable number of speakers have maximum scores for both conditions, making discrimination between conditions more difficult. We verified this explanation by running the same analyses

Table 8 F-scores after 40 and 175 training utterances for PATCOR

				Speakers									Average
				1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	
PATCOR	Hierarchical	$N = 40$	GMM 10	38	46	35	49	36	45	34	36	51	41.1
			GMM 80	40	43	38	47	37	43	33	38	46	40.6
			NMF	55	73	50	79	47	64	47	49	69	59.2
		$N = 175$	GMM 10	38	46	35	49	36	45	34	36	51	41.1
			GMM 80	54	71	44	70	46	58	40	43	62	54.2
			NMF	78	91	63	95	63	90	62	68	87	77.4
	Compositional	$N = 40$	GMM 10	41	47	35	49	36	45	34	36	50	41.4
			GMM 80	39	43	38	48	37	42	32	38	46	40.3
			NMF	53	66	45	79	46	63	42	49	66	56.6
		$N = 175$	GMM 10	41	47	35	49	36	45	34	36	50	41.4
			GMM 80	54	66	44	70	45	62	40	47	61	54.3
			NMF	72	83	61	95	62	87	58	69	85	74.7

The F-scores are interpolated using the LOWESS procedure.

for the overall lower GMM scores, using the same blocks, speech material and semantic structures. When comparing the flat and compositional frame structures, we found a considerable average improvement of 19% after one training block, $t_{(74)} = 9.8, p < 0.001$, and 7% for the maximal number of training blocks, $t_{(74)} = 3.6, p < 0.001$.

We probably obtain a good performance using a flat semantic structure, because the NMF-based acoustic representation is sufficiently distinctive to set each command apart. As a consequence, the more elaborated semantic frame structure becomes redundant. However, when the GMM-based processing flow provides less distinctive representations, information contained in the semantic frame structure becomes vital to the decision process. Nevertheless, overall results are in favour of the hierarchical approach, confirming our hypothesis that using additional knowledge in the form of a hierarchical semantic frame structure is an effective method to boost performance.

Conclusions

This work presents results on the recently recorded dysarthric speech database DOMOTICA-3, with speech intelligibility ranging from normal to severe dysarthric levels. Our NMF-based framework yields 90% to 100% F-score for all speakers, with typically 70% F-score after a single example. These scores validate the use of NMF-based learning as the basis for a self-taught vocal interface for normal and dysarthric speech.

The results on PATCOR and DOMOTICA-3 demonstrate higher asymptotic F-scores by using a more advanced semantic frame structure. The lower scores for the patience card game players, using words like "red" and "black" instead of anticipated semantic suit concepts, further confirm the importance of using a semantic structure with more levels similar to categories used in humans. However, the mismatch in user concepts and the concepts that designers had in mind in their applications is considered a weak aspect in our framework in spite of its overall strength. Therefore, we will focus on generic procedures in future work to induce a proper semantic structure. Moreover, further improvements are expected from embedding an algorithm to detect synonyms as alternative referents to the device slot values.

The hierarchical semantic frame structure was superimposed by a decision tree dominating decoded NMF activations. This decision stage can be integrated into the NMF procedure by using group sparsity [32] which obviates the need for a back-end decision stage in future work. All these moderations will boost performance, bringing us one step further in the design process towards a self-taught non-standard speech interface.

Endnote

[a] Note that frames without slots should have a corresponding entry in **a**, which determines the activation scores of the empty slot value.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

All authors made substantial contributions to conception and design, analysis and interpretation of the data, as well as drafting and revising the manuscript. All authors read and approved the final manuscript.

Acknowledgements

The research in this work is funded by IWT-SBO grant 100049.

Received: 31 January 2014 Accepted: 27 November 2014
Published online: 19 December 2014

References

1. G Zon, M Roerdink, *Using Voice to Control the Civil Flightdeck*. (Technical Report, NLR-TP-2006-720, National Aerospace Laboratory Amsterdam, Nederland, 200)
2. R Kuhn, J-C Junqua, P Nguyen, N Niedzielski, Rapid speaker adaptation in eigenvoice space. IEEE Trans. Speech Audio Process. **8**(6), 695–707 (2000)
3. G Potamianos, C Neti, in *AVSP 2001-International Conference on Auditory-Visual Speech Processing*. Automatic speechreading of impaired speech (Volterra, Italy, 2001), pp. 177–182
4. F Rudzicz, in *Proc SLPAT*. Acoustic transformations to improve the intelligibility of dysarthric speech (Association for Computational Linguistics Edinburgh, Scotland, 2011), pp. 11–21
5. P Green, J Carmichael, A Hatzis, P Enderby, MS Hawley, M Parker, in *Proc Interspeech*. Automatic speech recognition with sparse training data for dysarthric speakers (Geneva, Switzerland, 2003), pp. 1189–1192
6. M Parker, S Cunningham, P Enderby, M Hawley, P Green, Automatic speech recognition and training for severely dysarthric users of assistive technology: the stardust project. Clin. Linguist. Phon. **20**(2-3), 149–156 (2006)
7. AM Acrey, Speech recognition in individuals with dysarthria. PhD thesis, Texas Tech University (2012)
8. S-O Caballero-Morales, Estimation of phoneme-specific HMM topologies for the automatic recognition of dysarthric speech. Comput. Math. Methods Med. **2013** (2013). doi:10.1155/2013/297860
9. S-O Caballero-Morales, F Trujillo-Romero, Evolutionary approach for integration of multiple pronunciation patterns for enhancement of dysarthric speech recognition. Expert Syst. Appl. **41**(3), 841–852 (2014)
10. Y Hwang, D Shin, C-Y Yang, S-Y Lee, J Kim, B Kong, J Chung, S Kim, M Chung, in *Computers Helping People with Special Needs*. Lecture Notes in Computer Science, ed. by K Miesenberger, J Klaus, W Zagler, and A Karshmer. Developing a voice user interface with improved usability for people with dysarthria, vol. 7383 (Springer, Berlin Heidelberg, 2012), pp. 117–124
11. P Raghavendra, E Rosengren, S Hunnicutt, An investigation of different degrees of dysarthric speech as input to speaker-adaptive and speaker-dependent recognition systems. Augment. Altern. Commun. **17**(4), 265–275 (2001)
12. F Rudzicz, in *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility*. Assets '07. Comparing speaker-dependent and speaker-adaptive acoustic models for recognizing dysarthric speech (ACM New York, NY, USA, 2007), pp. 255–256. doi:10.1145/1296843.1296899 http://doi.acm.org/10.1145/1296843.1296899
13. M Matassoni, R Astudillo, A Natsamanis, M Ravanelli, in *Proc. Interspeech*. The dirha-grid corpus: baseline and tools for multi-room distant speech recognition using distributed microphones (Singapore, 2014), pp. 1613–1617
14. B Lecouteux, M Cacher, F Portet, in *Proc Interspeech*. Distant speech recognition in a smart home: comparison of several multisource ASRs in realistic conditions (Florence, Italy, 2011), pp. 2273–2276
15. H Christensen, I Casanuevo, S Cunningham, P Green, T Hain, in *Proc SLPAT*. homeService: voice-enabled assistive technology in the home using cloud-based automatic speech recognition (Grenoble, France, 2013), pp. 29–34
16. J Gemmeke, B Ons, M Tessema, J van de Loo, G De Pauw, W Daelemans, J Huyghe, J Derboven, L Vuegen, B Van Den Broeck, H Van hamme, in *Proc Interspeech*. Self-taught assistive vocal interfaces: an overview of the ALADIN project (Lyon, France, 2013), pp. 2038–2043
17. J Driesen, Discovering words in speech using matrix factorization. PhD thesis, K.U.Leuven, ESAT, July 2012
18. B Ons, N Tessema, J van de Loo, JF Gemmeke, in *Proc SLPAT*. A self learning vocal interface for speech-impaired users (Grenoble, France, 2013), pp. 1–9
19. Y Wang, A Acero, Rapid development of spoken language understanding grammars. Speech Commun. **48**(3-4), 390–416 (2006)
20. SB Davis, P Mermelstein, Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Trans. Acoustics Speech Signal Process. **28**(4), 357–366 (1980)
21. B Ons, JF Gemmeke, H Van hamme, Fast vocabulary acquisition in an NMF-based self-learning vocal user interface. Comput. Speech Lang. **28**(4), 997–1017 (2014)
22. H Van hamme, in *Proc. Interspeech*. HAC-models: a novel approach to continuous speech recognition (Brisbane, Australia, 2008), pp. 255–258
23. M Van Segbroeck, H Van hamme, Unsupervised learning of time-frequency patches as a noise-robust representation of speech. Speech Commun. **51**, 1124–1138 (2009)
24. A Cichocki, R Zdunek, A-H Phan, S Amari, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. (John Wiley & Sons, Ltd chichester, United Kingdom, 2009)
25. Z Akata, C Thurau, C Bauckhage, in *16th Computer Vision Winter Workshop*. Non-negative matrix factorization in multimodality data for segmentation and label prediction (Mitterberg, Austria, February 2–4 2011)
26. H Lee, J Yoo, S Choi, Semi-supervised nonnegative matrix factorization. Signal Process. Lett. IEEE. **17**(1), 4–7 (2010)
27. DD Lee, HS Seung, Learning the parts of objects by nonnegative matrix factorization. Nature. **401**, 788–791 (1999)
28. B Lize, D Katrien, FG Jort, H Van hamme, in *Proc SLPAT*. Comparing and combining classifiers for self-taught vocal interfaces (Grenoble, France, 2013), pp. 21–28
29. C Middag, Automatic analysis of pathological speech. PhD thesis. Ghent University, Belgium, 2012
30. WS Cleveland, Robust locally weighted regression and smoothing scatterplots. J. Am. Stat. Assoc. **74**(368), 829–836 (1979)
31. WS Cleveland, SJ Devlin, Locally weighted regression: an approach to regression analysis by local fitting. J. Am. Stat. Assoc. **83**(403), 596–610 (1988)
32. R Jaiswal, D Fitzgerald, E Coyle, S Rickard, in *Proc at 15th International Conference on Digital Audio Effects DAFx-12*. Shifted NMF with group sparsity for clustering NMF basis functions (York, UK, 2012), pp. 17–21

doi:10.1186/s13636-014-0043-4
Cite this article as: Ons *et al.*: The self-taught vocal interface. *EURASIP Journal on Audio, Speech, and Music Processing* 2014 **2014**:43.