A Self Learning Vocal Interface for Speech-impaired Users

Bart Ons[1]*, Netsanet Tessema*[1]*, Janneke van de Loo*[2]*, Jort F. Gemmeke*[1]*,*
Guy De Pauw[2]*, Walter Daelemans*[2]*, Hugo Van hamme*[1]

[1]ESAT, KU Leuven, Leuven, Belgium
[2]CLiPS - Computational Linguistics Group, University of Antwerp, Antwerp, Belgium
jort.gemmeke@esat.kuleuven.be

Abstract

In this work we describe research aimed at developing an assistive vocal interface for users with a speech impairment. In contrast to existing approaches, the vocal interface is self-learning, which means it is maximally adapted to the end-user and can be used with any language, dialect, vocabulary and grammar. The paper describes the overall learning framework and the vocabulary acquisition technique, and proposes a novel grammar induction technique based on weakly supervised hidden Markov model learning. We evaluate early implementations of these vocabulary and grammar learning components on two datasets: recorded sessions of a vocally guided card game by non-impaired speakers and speech-impaired users engaging in a home automation task.

Index Terms: vocal user interface, self-taught learning, dysarthric speech, non negative matrix factorization, hidden Markov models

1. Introduction

These days, vocal user interfaces (VUIs) allow us to control computers, smart phones, car navigation systems and domestic devices by voice. While still generally perceived as a luxury, assistive technology employing a VUI can make a prominent difference in the lives of individuals with a physical disability for whom operating and controlling devices would require exhaustive physical effort [1].

Unfortunately, even state-of-the-art speech recognition systems offer little, if any, robustness to dialectic or dysarthric speech (often encountered with disabled users), and are often restricted in their vocabulary and grammar. In practice, it is not feasible to design speech interfaces featuring custom acoustic and language models that cater to the dialectic and/or pathological speech of individual users, and adaptation of existing acoustic models is limited to only very mild speech pathologies [2, 3, 4, 5, 6]. Moreover, the user's voice may change over time due to progressive speech impairments.

Our aim is to build a VUI that is trained by the end-user himself, which means that it is maximally adapted to the — possibly dysarthric — speech of the user, and can be used with any vocabulary and grammar. The challenge is to learn both acoustics and grammar from a small number of examples, with as only supervisory information coarse annotation in the form of associated actions. For example, the annotation of the command "Turn on the television please", accompanied by a button press, would only be annotated at the utterance level with a device label (television) and an action label (turn on).

Our learning approach consists of two components that interact. Vocabulary acquisition first builds recurrent acoustic patterns representing words or parts of spoken commands, while grammar induction attempts to model the relationships between these patterns. For vocabulary acquisition, we build on existing work on child language learning modeling with non-negative matrix factorisation (NMF) [7]. For grammar induction, we propose the use of a weakly supervised Hidden Markov Model (HMM).

In short, we first use NMF to find recurrent acoustic patterns by mining utterance-level acoustic representations, supervised with relevant information about the action that was performed, such as a 'television' device and a 'turn on' action. Building on these, we then use the temporal occurrence of these patterns in the training data as observation features to train a multi-label version of a discrete HMM [8, 9]. In the HMM, the hidden states represent the collection of possible values in the data structures (devices and actions in the example). By mining the temporal occurrence of the NMF-based observations and the commonalities and differences across commands, the HMM is able to discover temporal structure in the commands, related to the data structures representing the actions.

The goals of our work are similar to those of [10, 11] in that we aim to discover acoustic patterns that recur in utterances and *ground* these by linking them to other modalities. However, to accommodate pathological voices, our work does not rely on pre-trained models, but they are learned from the speaker-specific acoustic data. In that sense, it shows similarities to the work in [12], but we learn form continuous speech and do not model low-level acoustics with an HMM. In terms of grammar learning, our task approaches unsupervised grammar induction [13, 14], but on a restricted domain with a small vocabulary.

We evaluate our learning framework on two databases: PATCOR, recorded sessions of a vocally guided card game by non-impaired speakers, and DOMOTICA-2, speech-impaired users engaging in a home automation task. The users were free to choose their own words and grammatical constructs to address the systems during the recording sessions.

The remainder of the paper is organised as follows. In section 2, we present an overview of the learning framework, describe the acoustic representations and introduce the NMF and HMM learning approaches. In section 3, the experimental setup is explained and in sections 4 and 5 the experimental results are presented and discussed. We conclude with our conclusions and thoughts for future work in section 6.

2. Architecture

2.1. Semantic frame representation of an utterance

A semantic *frame* is a data structure that contains all the relevant information (semantic concepts) associated with the ac-

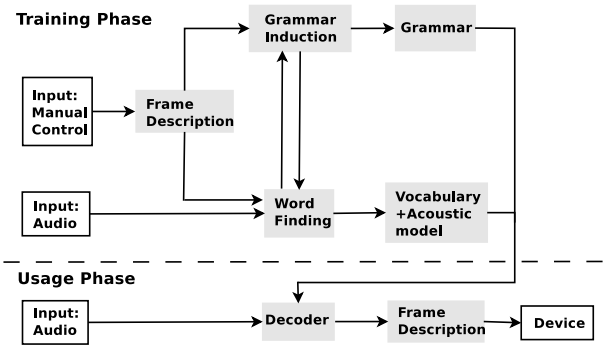

Figure 1: Overview of the vocal interface framework. The white boxes indicate events or systems outside the learning framework. The top panel shows the training phase and the bottom panel indicates the usage phase.

tion that is expressed in the spoken command. Semantic frames have been used in many spoken language processing applications [15]. A *frame* contains at least one *slot* representing a specific aspect of the *action*. Each *slot* in a frame can only be filled with a single *value*. A *frame description* of an *action* on the other hand, identifies a single frame out of the possible *frames* where the *action* is specified by the actual *slot values*.

2.2. The learning framework

The framework (Figure 1) is designed so it can learn from user interaction examples, i.e. a spoken command accompanied by an action on the device's user interface. For instance, users might say " Turn on the light" while pressing the button to switch on the light themselves or through the help of a care taker. The action performed on the device is translated into a *frame description*, which constitutes an abstraction layer making the learning algorithms application independent.

During the training phase, the word finding module looks for word-sized recurring acoustic patterns in the audio input that correlate well with the *frame description*. The *frame description* acts as a weak form of supervision in finding the recurring acoustic patterns. Here the term *weak supervision* is used because the supervision does not provide explicit information about the sequence of words within the spoken utterance.

The *grammar induction* module learns the relation between the different parts of a command. Given the frame description and the output of the word finding module, the grammar induction module learns the structure within commands, as well as the relation with the frame description during the training phase. During the usage phase, when only audio input is available, the grammar constrains the decoding process [16] and allows to propose a frame description of the spoken command. This frame description is then mapped onto an actual action on the device.

2.3. Audio representation

The word finding module in the training phase as well as the decoder in the usage phase need a suitable representation for the input speech. Both learning and recognition are based on NMF (section 2.4.1), which requires that the audio representation of an utterance be the sum of the representations of individual words. Therefore, and unlike main-stream ASR systems, an utterance is mapped to a vector of fixed size in three steps which are described below.

2.3.1. Spectral Representation

The first step of the audio processing chain extracts a 12-dimensional Mel Frequency Cepstral Coefficient (MFCC) representation of the short-term spectrum from speech segments of 25 ms with 10 ms overlap. The 12-dimensional MFCC is augmented with the log energy and the Δ and $\Delta\Delta$ features are appended, forming a 39 dimensional *spectral feature* stream.

2.3.2. Intermediate representations

The obtained MFCC spectral representations are further processed to form posteriorgrams from which the final representations described in section 2.3.3, are obtained. Two different forms of posteriorgrams are considered here: a *spectral feature* vector is either transformed into a vector of posterior probabilities of Gaussians forming a code book (soft VQ), or it is transformed to the posterior probability of phone classes.

In Soft Vector Quantisation, each *spectral feature* vector is softly assigned to all clusters in a code book. Each cluster is characterized by a Gaussian with full covariance. The degree of assignment is measured by the posterior probability of a Gaussian given the *spectral feature* vector.
The code book training starts off from a single cluster describing all training data. It is then split along the dominant eigenvector of its covariance matrix into two subclusters. The centres are refined with k-means iterations after which each subcluster is characterised by a full covariance Gaussian. This process is repeated, each time splitting the cluster with the largest volume as measured by the determinant of the covariance matrix. This process is either stopped when the desired number of clusters are obtained [17], which we will refer to by *Soft VQ*, or when the number of *spectral feature* vectors assigned to a cluster falls below a threshold, *minimum-number of frames*, which is referred to as *Adaptive Soft VQ*, because the number of clusters will depend on the amount of training data.

Phone posteriorgrams are constructed from 50 monophone HMMs (including a model for silence), each modeled by three states with GMM emission densities, connected in a strict left-to-right topology. The utterance is first transcribed into a phone lattice without using a phone-level language model. The acoustic likelihoods associated with the arcs are subsequently renormalised to posterior probabilities, which allows us to compute a posterior probability for each phone at any time.

A major difference with *Soft VQ* is that phone posteriorgrams exploit prior knowledge about the phone inventory that the user can produce.

2.3.3. Utterance-level HAC representation

The posteriorgrams of spectral feature clusters or of phone classes are not suitable to model directly with an NMF. To be able to discover recurring patterns in utterances, they need to be mapped to a representation of fixed dimension in which linearity holds, i.e. that the utterance-level speech representation is approximately equal to the sum of the speech representations

of the acoustic patterns it contains [18, 19]. A mapping that exhibits this property is the so-called histogram of acoustic co-occurrences (HAC) [19]. The HAC of a speech segment is the posterior joint probability of two *acoustic events* happening at a predefined time lag τ, accumulated over the entire segment. An acoustic event is the observation of a spectral feature vector from a particular cluster in the case of soft VQ, or the observation of a phone in the case of phone posteriorgrams. Since the HAC representation considers event pairs, its dimensionality is the square of the number of acoustic event classes. In this paper, we stack HAC vectors computed for multiple values of the time lag $\tau = 20, 50, 90$ and 200 ms into a single *augmented HAC vector* to characterise an utterance. When multiple (training) utterances are available, their augmented HAC representations are arranged as columns of a matrix $\mathbf{V}_a$.

2.4. Non-negative matrix factorisation

NMF uses non-negativity constraints for decomposing a matrix into its components [20, 21, 22, 23], i.e given a non-negative matrix $\mathbf{V}$ of size $[M\text{x}N]$, NMF approximately decomposes it into its non-negative components $\mathbf{W}$ of size $[M\text{x}R]$ and $\mathbf{H}$ of size $[R\text{x}N]$. Under the right conditions, NMF is able to find parts in data. In ASR, NMF is used to discover recurring acoustic patterns (word units) through some grounding information [24, 25, 26].

In this paper, we use the Kullback-Leibler divergence to quantify the approximation quality of the NMF as expressed in Eq 1.

$$(\mathbf{H}, \mathbf{W}) = \arg \min_{(\mathbf{H},\mathbf{W})} D_{KL}(\ \mathbf{V}\ ||\ [\mathbf{WH}]) \tag{1}$$

Finding the $\mathbf{W}$ and $\mathbf{H}$ that minimize this approximation metric for a given data matrix $\mathbf{V}$ is achieved using multiplicative update rules[20].

2.4.1. Supervised NMF word learning

To employ NMF for word learning, we use a weak form of supervision represented by $\mathbf{V}_g$, which is used together with the augmented HAC acoustic representation of all the training utterances stacked into a matrix $\mathbf{V}_a$. The supervision information links the discovered acoustic patterns to *slot values* and also helps NMF to avoid local optima of the Kullback-Leibler divergence. The supervision $\mathbf{V}_g$ is a label matrix where each column represents an utterance and each row represents a *slot value*. The presence of a *slot value* in an utterance is represented in the label matrix with a '1' and its absence with a '0'.

Through the factorization of the composite matrix constructed by vertical concatenation of $\mathbf{V}_g$ and $\mathbf{V}_a$, NMF discovers latent *slot value* representations in each column of $\mathbf{W}_a$. The columns of W_g link the learned acoustic patterns in columns of $\mathbf{W_a}$ to the *slot values* represented by the rows of $\mathbf{V}_g$. Furthermore, some extra columns of $\mathbf{W_a}$ and $\mathbf{W_g}$ are used to represent *filler words* (words which are present in the utterance but are not related to any *slot value*). The columns of $\mathbf{H}$ matrix indicate which columns of $\mathbf{W}_a$ and $\mathbf{W}_g$ are combined to reconstruct $\mathbf{V}_a$ and $\mathbf{V}_g$ respectively. The learned acoustic patterns in $\mathbf{W}_a$ and labeling information in $\mathbf{W}_g$ as given in Eq. 2 will be used in the testing phase to detect the learned acoustic units within unseen test utterances.

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

2.4.2. NMF in the usage phase

The learned NMF model is applied in two different approaches to decoding. Both decoders apply the learned NMF model to word-sized segments of speech in a *sliding window* analysis. A sliding window of a width of 300 ms and a shift of 100 ms is used to produce an *augmented HAC vector* at 100 ms intervals across an utterance. As a result, an utterance is represented by a matrix $\mathbf{V}_s$, containing one column per window position. By employing the NMF factorization Eq. 3, which is called the *local* NMF, the corresponding *slot value activations* are calculated.

$$\mathbf{H}_s = \arg \min_{\mathbf{H}_s} D_{KL}(\mathbf{V}_s || \mathbf{W}_a \mathbf{H}_s) \tag{3}$$

This is followed by the calculation of the activation matrix $\mathbf{A}_s$. Each column of the activation matrix contains labeling information of all *slot values* for a particular window position.

$$\mathbf{A}_s = \mathbf{W}_g \mathbf{H}_s \tag{4}$$

In the simplest form of decoding, called *NMF decoding*, the slot values are inferred directly from the local (sliding window) NMF. The activations for all slot values are accumulated over all window positions, i.e. over the complete utterance. Since each slot can have at most one value assigned, only the value hypothesis with the largest accumulated activation is kept per slot. The slot value is considered to be detected, only if the accumulated activation exceeds a threshold. The order in which the acoustic patterns related to the slot values occur in the utterance is therefore ignored. Since this procedure may result in multiple possible frames, we select the frame with the highest average probability mass.

In a refinement, called *HMM decoding*, the local NMF model generates a data stream which is modeled by an HMM. The HMM captures the relation between word usage – including word order – and frame descriptions of actions. Since the HMM models the sequential aspects of the utterance (such as word order), we consider the learning of this HMM a form of *grammar induction*. The details of this approach are explained in the next section.

2.5. Grammar induction

Identical or similar words (e.g. numbers) may refer to different slots, so slot-value pairs can only be assigned correctly from spoken input if grammar is taken into account. *HMM decoding* fixes the major shortcoming of *NMF decoding*, i.e. that the order in which slot values occur, is ignored. The local NMF stream is then modeled by an HMM, which is learned from the user interaction examples.

2.5.1. HMM learning

The activation sequence is modeled by a multi-labeling HMM [9]. Like in discrete-density HMMs, each state q is characterized by probabilities $b_j(q)$ over observations j. In this framework, the observation is characterized by a probability distribution derived from NMF atom activations, obtained as $\mathbf{H}_s$, normalized to sum to unity. The state probability is then

the inner product of this distribution with the state distribution.

Applied to this problem, each *semantic frame* is modeled by an HMM in which each *slot value* is assigned an HMM state referred to as *slot value state*. States are fully connected, with two exceptions. First, within slot transitions are prohibited, since each slot needs to be assigned only one value. Second, states can only transition to slot-value states within the same semantic frame, since each spoken command can only correspond to a single semantic frame. To limit the number of transition probabilities to be estimated, all transitions from states associated with a particular slot, to all states associated with another slot, share the same transition probability. The HMM will hence learn the sequence of slots in the user's utterances, but not the sequence of individual words. All the states can be initial or final states.

HMM training is done using the Baum-Welch algorithm [27]. Supervision information provided by the labeling matrix $\mathbf{V}_g$, is used to only assign non-zero state posteriors to *slot values* that are present in the *frame description* of an utterance. All non-zero entries of the state-transition matrix are initialised to (properly normalised) random values. The emission matrix is initialised by $\mathbf{W}_g$.

2.5.2. HMM decoding

During decoding, the maximum likelihood state sequence is obtained using the Viterbi algorithm for the given observation sequence $\mathbf{H}_s$. Visiting a state in an HMM corresponding to a semantic frame implies the corresponding *slot value* is detected. Since states representing slot values can only transition to states within the same semantic frame, the Viterbi search implicitly selects the most likely frame.

3. Experimental Setup

In this section, we give a description of the databases used for evaluation, the evaluation procedure and metrics.

3.1. Databases

3.1.1. PATCOR

The database PATCOR contains recordings of subjects playing a card game called "Patience" using spoken commands. The database contains 8 speakers with in total more than two thousand commands. The data was collected from unimpaired subjects with non-pathological speech, speaking Belgian Dutch. The users were free to choose their vocabulary and grammar, although in practice the vocabulary was limited indirectly by the number of cards, card positions and functionality.

A typical utterance in PATCOR is *"Put the four of clubs on the five of hearts"*. In this type of utterance, the order of the words plays a key role in discovering the utterance's meaning. The gold-standard frame descriptions of the utterances were created manually. In Table 1 an overview of the total number of frames, slots and slot values used is given. Since not all possible slot values occur for all speakers, Table 7 gives the actual number of slot values for each speaker. For a more detailed description of the frame descriptions that were used, as well as the slot values used for each speaker, we refer the reader to the technical report [28].

Table 1: *parameters of the speech databases*

Database	PATCOR	DOMOTICA
number of speakers	8	20
number of *frames*	2	4
number of *slots*	9	7
number of *slot values*	58	27
number of blocks	8	6

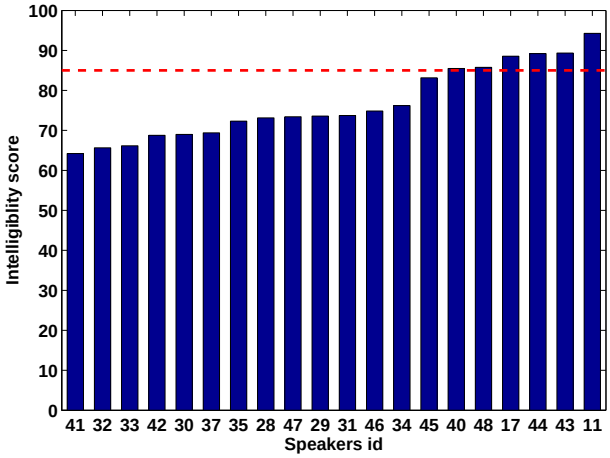

Figure 2: Speech intelligibility measurements of the speakers in DOMOTICA-2. The speakers are order by intelligibility score. Generally speaking, a score higher than 85% is non-pathological (see the dashed line).

3.1.2. DOMOTICA-2

The DOMOTICA-2 database contains recordings of impaired, dysarthric speakers controlling a home automation system. A typical DOMOTICA-2 utterance would look like "*Turn on the kitchen light*".

Since collecting a large number of realistic, spontaneous spoken commands is difficult due to the targeted users getting tired quickly, a two-phase data collection method was used. In the first phase, 9 users were asked to control 31 different appliances in a 3D environment [28], guided by a visualised scenario in order to ensure an unbiased choice of words and grammar. In the second phase, these command lists were read back repeatedly by 21 test users. Of these 21, 8 speakers were selected based on their increased risk for degenerate voice rather than currently having a pathological voice.

For all speakers, speech intelligibility scores were obtained by analysing their recorded speech using an automated tool [29]. These scores are shown in Fig. 2. Table 1 gives an overview of the total number of frames, slots and slot values. For some speakers some slot values were not used, since some commands were not spoken enough times to allow a meaningful evaluation; Table 7 gives the actual number of slot values for each speaker. For a more detailed description of the slot values used for each speaker we refer the reader to the technical report [28].

3.2. Methodology

The goal of the experiments is to evaluate the performance as a function of the amount of training data used. However, since

this means the amount of training data can be very small, a form of cross validation is needed to obtain statistically meaningful scores.

First, we divide the spoken commands (utterances) of each speaker into equal or nearly equal parts called *blocks*. The k blocks are created by minimising the Jensen-Shannon divergence (JSD) between the slot value distributions of all blocks. This optimisation is performed in an iterative process starting by dividing all utterances randomly into k blocks and then swapping at each iteration those two utterances that minimise the JSD the most from one block to the other one. The process stops when the JSD is minimised, i.e. when there are no swaps left that can lower the JSD. The slot values are then approximately evenly distributed throughout the blocks. Under the constraint that each slot value should occur at least once in each block, some slot values are excluded from the frame structure, meaning that the spoken words corresponding to these slot values, become filler words: they are not supervised and they are not scored anymore. Such adaptation to the supervision is speaker dependent and the number of slot values used for each speaker can be found in [28]. Utterances without any slot values were removed from the training and test sets.

To evaluate the learning speed of our framework, we created a $k \times k$ latin square in which each block occurs exactly once in each row and in each column. We selected five rows of the latin square to create a five-fold cross-validation experiment in which the train and test sets respectively increase and decrease in size. In each fold, we start with an experiment where only one block is used for training while the remaining $k-1$ blocks are used for testing. We incrementally increase the number of blocks n used for training in the subsequent experiments and the last experiment will be performed with $n = k-1$ training blocks and one test block. Throughout the folds, the train and test sets are always composed of different blocks allowing for a more reliable scoring.

3.3. Parameters

The number of frames needed to have a reliable estimation of the cluster centres, depends on the dimensionality of the feature vectors. The minimum number of frames used for adaptive codebook training is chosen to be 78, two times the dimensionality of the MFCC feature vectors. For PATCOR, the resulting VQ codebook sizes typically ranged from 40 for the smallest training set to 145 for the largest training set. For DOMOTICA-2, the resulting codebook sizes typically ranged from 36 for the smallest training set to 118 for the largest training set.

For both databases, phone posteriorgrams were obtained using a free phone recognizer using a unigram language model. The phone recognizer was trained on a dataset containing recordings of selected radio and television news broadcasts in the same language as the collected databases. Phones are modeled with 3-state HMMs and in total 48845 tied Gaussians are used in the acoustic model. The phonetic alphabet includes one noise unit and one silence unit in addition to 48 phones.

For the utterance-based HAC representations, from both VQ and phone posteriorgrams, only the top-three largest indices at each time frame were retained.

3.4. Evaluation

For each utterance in the databases, we have a manually constructed gold standard frame description, which is used as a reference for system evaluation. In this reference frame description, the slot values that are expressed in the utterance, are filled in. The system was evaluated by comparing the automatically induced frame descriptions to the gold standard reference frames. The used metric is the *slot $F_{\beta=1}$-score*, which is the harmonic mean of the slot precision and the slot recall. These metrics are commonly used for the evaluation of frame-based systems for spoken language understanding [15]. The following formulas were used for calculation:

$$\text{slot precision} = \frac{\text{\# correctly filled slots}}{\text{\# total filled slots in induced frame}} \quad (5)$$

$$\text{slot recall} = \frac{\text{\# correctly filled slots}}{\text{\# total filled slots in reference frame}} \quad (6)$$

$$\text{slot } F_{\beta=1}\text{-score} = 2 \cdot \frac{\text{slot precision} \cdot \text{slot recall}}{\text{slot precision} + \text{slot recall}} \quad (7)$$

This means that only slots that are filled with a *correct* value are rewarded, and both slots that are falsely filled and slots that are falsely left empty are penalised. When an induced frame is of another type than the corresponding reference frame, the filled slots in the induced frame and in the reference frame are consequently different, which automatically results in a relatively large drop in the slot F-score. It should be noted that the reported F-scores aggregate slot counts over all five folds.

4. Results

In Fig. 3, F-scores for eight speakers per database are depicted as a function of the average number of utterances in the training set. The F-scores against increasing train set sizes provides some insight into the self-learning aspect of the framework. For each database, there are two graphs, one graph depicting NMF learning of slot value representations and one graph depicting HMM-based grammar induction.

For visibility, Fig. 3 does not contain all speakers from DOMOTICA-2. For this dataset, all F-scores for the NMF-based word finding module are presented in Table 2 and all scores for the HMM-based grammar induction module are presented in Table 3. There is one column for each speaker and the rows indicate the number of blocks in the training sets.

4.1. PATCOR

When we compare the respective F-scores for each speaker and for each training set size, we find a significant difference between the scores of the word finding module and the grammar induction module using a paired student's t-test, $t(55) = 5, 11$, $p < 0.001$. On average, the grammar induction module improves the F-score with 5%, but the improvement varies between speakers. For some speakers, the induced grammar provides a considerable improvement, for instance for speaker 3, The improvement is 16% on average, $t(6) = 33, 16$, $p < 0.001$. However, for instance, for speaker 5, we don't find a substantial improvement using the grammar induction module. In any case, using grammar induction does not seem to degrade the performance for any user in PATCOR.

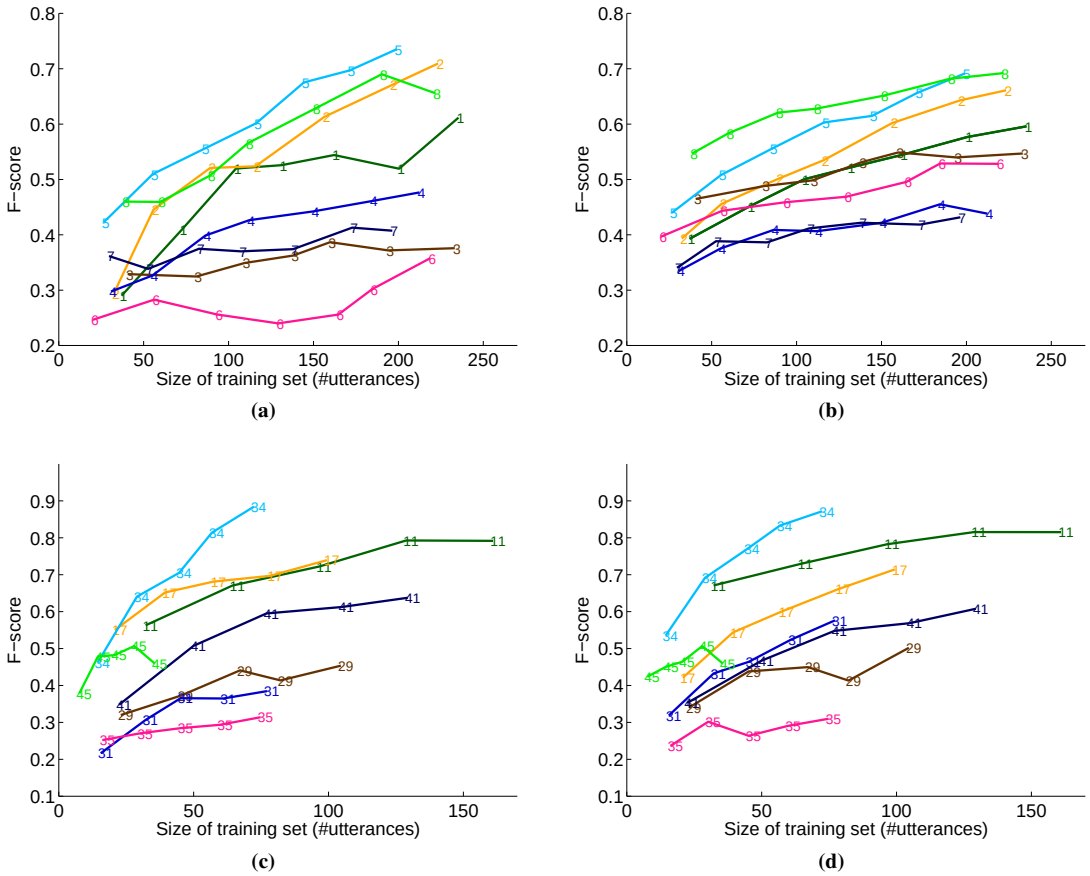

Figure 3: The F-scores per speaker against the averaged number of utterances in the respective training sets. In Panel (a), the NMF-based results of the word finding module for the PATCOR database are depicted. In Panel (b), the results of the word finding module augmented with the HMM-based grammar induction tool are displayed. Panel (c) and Panel (d) display the same results as Panel (a) and Panel (b), respectively, for eight selected speakers in the DOMOTICA-2 database.

Table 2: *F-scores for NMF word learning for all speakers of* DOMOTICA-2 *and all training set sizes*

speaker	11	17	28	29	30	31	32	33	34	35	37	40	41	42	43	44	45	46	47	48
1 block	0.56	0.55	0.30	0.32	0.27	0.22	0.22	0.27	0.46	0.25	0.24	0.36	0.35	0.28	0.34	0.31	0.38	0.16	0.17	0.36
2 blocks	0.67	0.65	0.36	0.37	0.32	0.31	0.26	0.31	0.64	0.27	0.30	0.44	0.51	0.29	0.37	0.46	0.48	0.16	0.17	0.29
3 blocks	0.72	0.68	0.41	0.44	0.40	0.37	0.33	0.32	0.71	0.28	0.36	0.50	0.60	0.32	0.39	0.53	0.48	0.20	0.15	0.33
4 blocks	0.79	0.70	0.50	0.41	0.41	0.36	0.32	0.32	0.81	0.29	0.36	0.48	0.61	0.41	0.38	0.61	0.51	0.17	0.14	0.29
5 blocks	0.79	0.74	0.48	0.45	0.43	0.38	0.38	0.40	0.88	0.31	0.44	0.53	0.64	0.45	0.44	0.63	0.46	0.22	0.13	0.26

Table 3: *F-scores for HMM grammar induction for all speakers of* DOMOTICA-2 *and all training set sizes*

speaker	11	17	28	29	30	31	32	33	34	35	37	40	41	42	43	44	45	46	47	48
1 block	0.67	0.42	0.32	0.34	0.26	0.32	0.21	0.18	0.54	0.24	0.27	0.33	0.35	0.21	0.32	0.43	0.42	0.20	0.18	0.40
2 blocks	0.73	0.54	0.36	0.44	0.36	0.43	0.24	0.31	0.69	0.30	0.28	0.45	0.47	0.26	0.44	0.55	0.45	0.23	0.19	0.46
3 blocks	0.78	0.60	0.43	0.45	0.46	0.46	0.25	0.29	0.77	0.26	0.45	0.58	0.55	0.33	0.41	0.58	0.46	0.25	0.16	0.48
4 blocks	0.82	0.66	0.49	0.41	0.50	0.52	0.31	0.32	0.83	0.29	0.37	0.59	0.57	0.31	0.32	0.66	0.51	0.22	0.19	0.58
5 blocks	0.82	0.71	0.48	0.50	0.43	0.58	0.29	0.35	0.87	0.31	0.50	0.60	0.61	0.28	0.59	0.70	0.46	0.24	0.19	0.61

Results are in the same range as the reported word finding results in [25], however, there are some speaker dependent differences in performance due to different experimental settings. The major discrepancies in settings are scoring and grammar discovery. While we report F-scores and investigate automatically induced grammar structures in this study, slot value recall scores are reported in [25] and frame decoding is guided by a handcrafted grammar. Additionally, the feature representations are also different between the two studies. While we combine phone posteriorgrams and adaptive softVQ for building the acoustic feature representations, the feature representation is based on softVQ using more larger codebooks in [25].

4.2. Domotica-2

For DOMOTICA-2, we find a small but significant improvement using a paired student's t-test when comparing the F-scores between the word finding module and the grammar induction module for each speaker and training set size (see Fig. 3c and Fig. 3d), $t(99) = 3, 24$, $p < 0.01$. On average the grammar induction module cause an increase in F-scores of about 3%. For some speakers, the F-score improvements were more pronounced than for others. For instance, F-scores for speaker 31 improved on average with a decimal of 0.14, $t(4) = 7, 6$, $p < 0.05$ while the F-scores for speaker 17 decreased with 8%, $t(4) = -3.77, p < 0.05$.

The differences between speakers is related to the intelligibility scores. We found a significant Kandall's tau rank correlation equal to $0.41, p < 0.05$ for the average F-score per speaker and their respective intelligibility score. There are trend lines in Fig. 3c and Fig. 3d that are rather short because the amount of data was limited, such as the graphs for speaker 35, resulting from early fatigue for some speakers in the recording phase of the DOMOTICA-2 corpus.

5. Discussion

In the word finding module, we aim to find the acoustic representation of the words corresponding to slot values in a semantic frame. In the grammar induction tool, the temporal structure in the commands is discovered and related to the semantic frame structure of the spoken commands. Positive scores necessitate a positive evaluation on both aspects, that is the correct recognition of the spoken words and the correct allocation of the recognised words to the slots in the semantic frame structure. The second aspect is not a trivial issue for the utterances used in the PATCOR database. For instance, in the utterance "Put the four of clubs on the five of hearts", words like "four" and "clubs" are related to the moving card while the same words are sometimes used to define the destination of the move. Some speakers specify the moving card first while others may specify the destination card first. Although spoken words are sometimes identical, different slot value labels specify different meanings. It can be seen in Fig 3b that the VUI gradually succeeds to distinguish these slots corresponding to the moved card and the destination card for at least some speakers, such as speaker $2, 5$ and 8. Scores above 0.5 are only possible when the correct slots are recognized, such as the slots related to the moving card versus the slots related to the destination card in PATCOR.

The NMF-based word finding module is able to learn more than words, as some context information of the words is incorporated in the slot value representations. The features used in NMF learning consist of the co-occurrence of acoustic events over multiple delays, up to $\tau = 200$ ms, allowing for learning context over spoken word boundaries. Moreover, the learned context of a word also involves the co-occurrence of acoustic events with the frame slot events of the demonstrated commands. The learned context is helpful in identifying the words but also the frame slots for some speakers as can be seen in Fig 3a and Fig 3c. However, context learning in NMF over word boundaries is only possible in a local time context because co-occurrence of acoustic events over longer time delays are more divergent. Useful time delays might be extended by using probabilistic time delays instead of fixed ones used here. The use of longer time delays in learning the co-occurrence of events poses a challenge for future research.

Although the frame structure is acquired for some speakers without using the grammar induction module, not all speakers display good scores without grammar induction, such as speakers $3, 6$ and 7 in PATCOR. The HMM-based grammar induction tool improves the learning of the frame structure, especially for those speakers for which the NMF word finding module demonstrates insufficiencies. The results demonstrated in Fig. 3 and Table 3 are encouraging in the sense that the graphs of all speakers in Fig. 3 tend to rise by increasing training set sizes, demonstrating the self-learning ability of the investigated framework. Further directions of research includes the acceleration of the learning plots for normal and dysarthric speech. Accelerating the speed of learning is especially important for speech-impaired users, because they have to make more effort to utter commands and train the system. Besides accelerating the speed of learning, it remains an open issue at which level the scores tend to level off. Obviously, all scores presented in the graphs of Fig. 3b and Fig. 3d are not at levelling off for the largest training set, as the training data is too scarce for the self learning VUI to reach maximal performance. More data is needed to find out the maximal performance of the system and the relation between maximal scores and intelligibility of the users. We could help this issue by gathering more data or by sharing the emission probabilities for particular slot values sharing identical words similar to the sharing of the transition probabilities explained in Section 2.5.1.

There are some differences in performance between databases. Our framework performs best for intelligible speech. The speakers with higher F-scores for the DOMOTICA-2 database are the speakers with the higher intelligibility scores close to 85%. The performance of our framework for different speakers in DOMOTICA-2 demonstrates a larger variability and more spurious trajectories in Fig.3d than for normal speakers in PATCOR. Low scores are corresponding to a low number of slot values which in turn is corresponding to a limited number of recorded utterances due to early fatigue. However, the scores between the two databases are difficult to compare since the complexities of the categorical decisions are different from each other. For instance, there are more frame slot values per slot in PATCOR than in DOMOTICA-2 and there is more hierarchical structure in the PATCOR-commands compared to the DOMOTICA-2-commands, making the recognition of PATCOR-commands much more difficult. In future research, we will evaluate our framework on more databases allowing us to compare the strengths and weaknesses of our system with other small-vocabulary, speaker-dependent systems, such as those described in [2, 6].

6. Conclusion

In this work we described research aimed at developing an assistive vocal interface for people with a speech impairment.

In contrast to existing approaches, the vocal interface is self-learning which means it is maximally adapted to the end-user and can be used with any language, dialect, vocabulary and grammar. We proposed a novel grammar induction technique, based on weakly supervised HMM learning, and we evaluated early implementations of these vocabulary and grammar learning components on two datasets: recorded sessions of a vocally guided card game by non-impaired speakers, and speech-impaired users engaging in a home automation task.

While the performance varied widely between speakers, both for impaired and non-impaired speakers, performance did improve even with relatively small amounts of additional training data. This demonstrates the potential of the self-learning vocal interface. Additionally, the proposed HMM approach to weakly supervised grammar induction did improve the results for all but a few speakers, indicating that a limited form of grammar induction is not only feasible, but also beneficial to distinguish between commands. Future work will focus not only on a detailed analysis of the obtained results, such as the grammars that were inferred and the relation between speech pathology and performance, but also on improvements such as more advanced acoustic modelling techniques, hierarchical approaches of HMM learning, and integrating grammar induction and vocabulary acquisition in a single probabilistic framework.

7. Appendix

Table 4: *number of* slot values *and maximum codebook size*

PATCOR		
speaker id	**number of *slot values***	**maximum codebook size**
1	29	117
2	37	145
3	23	152
4	27	78
5	25	151
6	18	189
7	27	165
8	19	142

DOMOTICA-2		
speaker id	**number of *slot values***	**maximum codebook size**
11	22	149
17	18	81
28	18	115
29	9	138
30	14	94
31	12	52
32	17	200
33	11	93
34	6	59
35	13	187
37	13	94
40	18	126
41	18	169
42	17	87
43	4	62
44	18	78
45	3	63
46	19	164
47	17	135
48	5	79

8. Acknowledgements

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

9. References

[1] J. Noyes and C. Frankish, "Speech recognition technology for individuals with disabilities," *Augmentative and Alternative Communication*, vol. 8, no. 4, pp. 297–303, 1992.

[2] H. Christensen, S. Cunningham, C. Fox, P. Green, and T. Hain, "A comparative study of adaptive, automatic recognition of disordered speech," in *Proc Interspeech 2012*, Portland, Oregon, US, Sep 2012.

[3] K. T. Mengistu and F. Rudzicz, "Comparing humans and automatic speech recognition systems in recognizing dysarthric speech," in *Proceedings of the Canadian Conference on Artificial Intelligence*, 2011.

[4] H. V. Sharma and M. Hasegawa-Johnson, "State transition interpolation and map adaptation for hmm-based dysarthric speech recognition," in *HLT/NAACL Workshop on Speech and Language Processing for Assistive Technology (SLPAT)*, 2010, pp. 72–79.

[5] F. Rudzicz, "Acoustic transformations to improve the intelligibility of dysarthric speech," in *Proceedings of the Second Workshop on Speech and Language Processing for Assistive Technologies (SLPAT2011)*, 2011.

[6] M. S. Hawley, P. Enderby, P. Green, S. Cunningham, S. Brownsell, J. Carmichael, M. Parker, A. Hatzis, P. O'Neill, and R. Palmer, "A speech-controlled environmental control system for people with severe dysarthria," *Medical Engineering & Physics*, vol. 5, no. 29, pp. 586 – 593, 2007.

[7] J. Driesen, J. Gemmeke, and H. Van hamme, "Weakly supervised keyword learning using sparse representations of speech," in *Proceedings of the 36th International Conference on Acoustics, Speech and Signal Processing*, Kyoto, Japan, 2012.

[8] M. Nishimura and K. Toshioka, "Hmm-based speech recognition using multi-dimensional multi-labeling," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '87.*, vol. 12, 1987, pp. 1163–1166.

[9] J. Hernando, l.B. Maririo, A. Moreno, and C. Nadeu, "Multiple multilabeling applied to hmm-based noisy speech recognition," in *Proc. ICSP '93*, 1993.

[10] R. Taguchi, N. Iwahashi, K. Funakoshi, M. Nakano, T. Nose, and T. Nitta, *Human Machine Interaction - Getting Closer*. InTech, 2012, ch. Learning Physically Grounded Lexicons from Spoken Utterances.

[11] D. Roy, "Grounded spoken language acquisition: Experiments in word learning," *IEEE Transactions on Multimedia*, vol. 5(2), pp. 197–209, 2003.

[12] I. Ayllon Clemente, M. Heckmann, and B. Wrede, "Incremental word learning: Efficient hmm initialization and large margin discriminative adaptation," *Speech Communication*, vol. 54, pp. 1029–1048, Nov. 2012.

[13] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, Aug 2011. [Online]. Available: http://leon.bottou.org/papers/collobert-2011

[14] D. Klein, "The unsupervised learning of natural language structure," Ph.D. dissertation, Stanford University, 2005.

[15] Y. Wang, L. Deng, and A. Acero, "Semantic frame-based spoken language understanding," in *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, G. Tur and R. D. Mori, Eds. West-Sussex, UK: Wiley, 2011, ch. 3, pp. 41–91.

[16] J. van de Loo, G. De Pauw, J. Gemmeke, P. Karsmakers, B. Van Den Broeck, W. Daelemans, and H. Van hamme, "Towards shallow grammar induction for an adaptive assistive vocal interface: a concept tagging approach," in *Proceedings NLP4ITA*, 2012, pp. 27–34.

[17] F. Class, A. Kaltenmeir, P. Regal-Brietzmann, and K. Trottler, "Fast speaker adaptation combined with soft vector quantization in an hmm speech recognition system," in *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, vol. 1, 1992, pp. 461–464 vol.1.

[18] J. Driesen and H. Van hamme, "Fast word acquisition in an NMF-based learning framework," in *Proceedings of the 36th International Conference on Acoustics, Speech and Signal Processing*, Kyoto, Japan, 2012.

[19] H. Van hamme, "Hac-models: a novel approach to continuous speech recognition," in *Proceedings INTERSPEECH*, 2008, pp. 2554–2557.

[20] D. Lee and H. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[21] J. Eggert and E. Korner, "Sparse coding and nmf," in *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, vol. 4, 2004, pp. 2529–2533 vol.4.

[22] Y.-X. Wang and Y.-J. Zhang, "Nonnegative matrix factorization: A comprehensive review," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 25, no. 6, pp. 1336–1353, 2013.

[23] H. Lee, J. Yoo, and S. Choi, "Semi-supervised nonnegative matrix factorization," *Signal Processing Letters, IEEE*, vol. 17, no. 1, pp. 4–7, 2010.

[24] L. Boves, L. ten Bosch, and R. Moore, "Acorns-towards computational modeling of communication and recognition skills," in *Proc. IEEE int. Conf. On Cognitive informatics*, California, USA, 2007, pp. 349–355.

[25] J. F. Gemmeke, J. van de Loo, G. De Pauw, J. Driesen, H. Van hamme, and W. Daelemans, "A self-learning assistive vocal interface based on vocabulary learning and grammar induction," in *Proc. INTERSPEECH*, 2012, pp. 1–4.

[26] B. Ons, J. F. Gemmeke, and H. Van hamme, "Label noise robustness and learning speed in a self-learning vocal user interface," in *Proc. of the International Workshop on Spoken Dialog Systems (IWSDS)*, Ermenonville, France, 2012.

[27] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[28] N. Tessema, B. Ons, J. Gemmeke, and H. Van hamme, "Technical report (aladin-tr01)," KULeuven ESAT-PSI, Tech. Rep., 2013.

[29] C. Middag, "Automatic analysis of pathological speech," Ph.D. dissertation, Ghent University, Belgium, 2012.