Machine learning techniques for semantic analysis of dysarthric speech: An experimental study

Vladimir Despotovic⁎,a, Oliver Walterb, Reinhold Haeb-Umbachb

a Technical Faculty of Bor, University of Belgrade, Bor, Serbia
b Department of Communications Engineering, University of Paderborn, Paderborn, Germany

A R T I C L E   I N F O

Keywords:
Semantic analysis
Spoken language understanding
Machine learning
Dysarthric speech
Acoustic units

A B S T R A C T

We present an experimental comparison of seven state-of-the-art machine learning algorithms for the task of semantic analysis of spoken input, with a special emphasis on applications for dysarthric speech. Dysarthria is a motor speech disorder, which is characterized by poor articulation of phonemes. In order to cater for these non-canonical phoneme realizations, we employed an unsupervised learning approach to estimate the acoustic models for speech recognition, which does not require a literal transcription of the training data. Even for the subsequent task of semantic analysis, only weak supervision is employed, whereby the training utterance is accompanied by a semantic label only, rather than a literal transcription. Results on two databases, one of them containing dysarthric speech, are presented showing that Markov logic networks and conditional random fields substantially outperform other machine learning approaches. Markov logic networks have proved to be especially robust to recognition errors, which are caused by imprecise articulation in dysarthric speech.

1. Introduction

Semantic analysis is the task of learning the mapping of spoken language to a semantic representation, and thus discovering the meaning of an utterance. Designing a meaning representation to express the spoken language is not a trivial task. Early approaches used first-order or higher-order logic to represent meanings (Mori et al., 2008; Montague, 1970). One of the approaches that is widely used in natural language processing (NLP) is based on semantic frames. Semantic frames are composed of slots, which represent specific attributes of the spoken utterance. The task here is three-fold: i) Target word detection finds semantically relevant words in an utterance (Coppola et al., 2009); ii) frame classification determines the frame that corresponds to an action or a domain of interest; iii) slot filling finds the slot-values that correspond to frame attributes of the input utterance (Wang, 2010). A more recent trend is the use of distributional semantics, where meanings of the words are determined based on the context in which they occur. In this way word meaning can be extracted from text (or speech) corpora on a large scale. Contexts are represented using vectors of frequencies of other words co-occurring with a word being modelled (Lenci, 2008). To represent the meaning of entire utterances a compositional model is used which composes the vectors for the words contained in an utterance to create a vector representation of the utterance (Bellegarda and Monz, 2016; Kartsaklis, 2014). This approach, however, hinges on the availability of large training corpora, which are usually not available for applications with dysarthric speech. We therefore decided to use the semantic frame representation of the meanings in our work since it is more appropriate for a domain specific task with limited available training data.

While semantic analysis in NLP assumes the processing of typed input (written language), we are interested in determining the meaning of spoken language here. This poses additional challenges, since we also have to deal with noise and the inaccuracies of automatic speech recognition (ASR). A straightforward way to solve this problem is to use a word-based ASR system that transforms spoken input into word sequences, and then apply the techniques already developed for processing written language. However, spoken language often does not follow the grammar and the syntactic structure of the written language, and is rather spontaneous, involving self-corrections, repetitions, and other irregularities. Moreover, ASR is error-prone and it outputs word sequences with no structure information (e.g. interpolation). Therefore it is necessary to adapt the natural language semantic analyser to cope with the problems of spoken language (Despotovic et al., 2015). An excellent survey of techniques for integrating ASR and spoken language understanding can be found in (Mori et al., 2008).

We are especially interested in building a semantic analyser that can be used with dysarthric speech, which is poorly articulated and often hardly intelligible. Dysarthria is a motor speech disorder caused by
problems controlling the muscles used in speech production. It is characterized by uneven speech rhythm and volume, slow, weak or slurred speech that is difficult to understand for listeners unfamiliar with the particular speech disorder (Christensen et al., 2013). Common causes of dysarthria include neurological disorders such as stroke, brain trauma, brain tumors, amyotrophic lateral sclerosis, cerebral palsy, multiple sclerosis, Parkinson’s disease, surgery or weakness of the tongue. It is often accompanied by severe physical impairments that make standard access to other devices (e.g. keyboard, mouse, touchscreen, adaptive pointing device) used in computer based assistive technology inefficient. ASR can, therefore, help individuals with dysarthria to interact with their environment. Unfortunately, due to the variability of their articulatory output, the use of standard speaker-independent ASR systems is not possible (Mengistu and Rudzicz, 2011). Furthermore, word boundaries in dysarthric speech are less apparent than in normal speech, which prohibits automatic recognition of the string of words. The experience is similar to listening to someone speaking a foreign language (Lansford et al., 2011). Hence, we propose an approach where we bypass word segmentation and try to learn a semantic analyser directly from the recognized subword unit sequence. Furthermore, to accommodate for the deviation of dysarthric speech from standard pronunciation, the sub-word unit representations are automatically learned from the speech input. This avoids the need for a custom pronunciation lexicon for each speaker uttering dysarthric speech. The subword units are discovered as acoustic segments that have been consistently observed in training data. These units we obtain in an unsupervised way, in the absence of the labelled training data or a pronunciation lexicon (Walter et al., 2013), which is very important, as it is inherently difficult to obtain labelled training data for speakers with dysarthria. Hence, unsupervised methods might be of particular interest. Also, this potentially allows for an unlimited vocabulary. In that sense, our task is similar to Gaspers and Cimiano (2014) where a semantic parser is learned from a sequence of phonemes at the output of the phoneme recognizer, which are subsequently segmented into (sub) word-like units.

Our aim in this paper is to give a comprehensive comparative analysis of different machine learning approaches for the task of semantic analysis of dysarthric speech, although we present the results for normal speaking users as well. We use multinomial naive Bayes (MNB), support vector machines (SVM), maximum entropy (MaxEnt), linear discriminant analysis (LDA), non-negative matrix factorization (NMF), conditional random fields (CRF) and Markov logic networks (MLN). Naive Bayes is commonly used in text classification due to its simplicity and low complexity (Kilimci and Ganiz, 2015). There are known applications in spoken language understanding (SLU), e.g. for a task classification in the context of a public transport information dialog system in Chinese language (Weilin et al., 2003). Tur and De Mori compare the performance of naive Bayes and SVM and show that SVM remains robust even when the dimensionality of the problem increases, while naive Bayes is preferred where statistical estimation does not suffer from the curse of dimensionality (Tur and De Mori, 2011). Deoras et al. propose a joint decoding of words and semantic tags for SLU where the optimal word and the semantic slot sequence are predicted jointly given the input acoustic stream, instead of employing a cascade approach, where the output of ASR is fed into the semantic analyser. These statistical models are trained individually for both steps. For the joint decoding task MaxEnt and CRF models have similar performance, while CRF slightly outperforms MaxEnt for the cascade approach (Deoras et al., 2013). Wang and Acero show that linear-chain conditional random fields (CRF) perform best among several discriminative models when converting the SLU problem into a sequential labelling task (Wang and Acero, 2006). A major disadvantage of these discriminative models is the necessity of labelling the training utterances with semantic representations at the word-level (Mairese et al., 2009). Wuwiriwatchai and Furui compare the results on confidence scoring for a spoken dialogue system for Thai language using the Fisher LDA and SVMs with linear, polynomial and radial basis function kernels, concluding that all three SVMs outperform LDA for a given task (Wuwiriwatchai and Furui, 2003). Ons, Gemieke and Van hamme propose an MLN-based vocal-user interface used in a home automation system for speakers with dysarthria to find recurrent acoustic and semantic patterns corresponding to spoken commands (Ons et al., 2014). Kennington and Schlangen use MLNs for situated incremental natural language understanding from the noisy input, coming from the output of the ASR (Kennington and Schlangen, 2014). Khot et al. use MLNs for automatic question answering in standardized science exams (Khot et al., 2015). Despotovic, Walter and Haeb-Umbach apply MLNs to the semantic analysis of spoken input and gain significantly better results compared to NMF, SVM and MNB based approaches for both normal speaking and dysarthric users (Despotovic et al., 2015).

The current paper presents an extension of our work in Despotovic et al. (2015): We employ more machine learning algorithms in our comparison, and we present the experimental results in more detail. Furthermore, the tested machine learning approaches are described in more detail. All the experiments were performed in two domains, a home automation task for speech impaired people (DOM-OTICA 3) and a vocally guided card game named patience (PATCOR), containing speech of normal-speaking persons (Gemmeke et al., 2013). For both domains subjects were giving commands during the training phase freely, not restricted to any particular words or grammatical constructs. For the mapping task only a weak supervision was required, since only the actions were annotated using semantic frames, not the exact words that were used to express the command. Since we are particularly interested in dysarthric speech, which is often characterized by non-canonical phoneme realizations, we employ models of acoustic units that are learned speaker-dependently in an unsupervised fashion rather than using a speaker-independent phoneme recognizer.

The remainder of the paper is organized as follows. Section 2 presents details of acoustic pre-processing. Section 3 gives a brief overview of machine learning algorithms tested in this paper. The speech corpora and evaluation procedure are presented in Section 4. Results and discussion are described in Section 5, followed by concluding remarks in Section 6.

2. Acoustic preprocessing

Acoustic pre-processing is the task of partitioning an input stream of speech and deriving a set of parameters to represent speech in a form which is suitable for subsequent processing (Singh et al., 2012). An adequate acoustic representation is especially important for dysarthric speech, where the speech rate is reduced, vowels may be distorted and word boundaries are less apparent. Moreover, an increase in phoneme transition duration and in syllable and word duration is observed (Duffy, 2012). Details of acoustic representation and feature extraction are given in this section.

2.1. Acoustic representation

In order to learn the mappings to semantic representations directly from the raw speech, we employ an intermediate acoustic representation of the spoken input in terms of acoustic unit descriptors (AUD), which are subword units learned in an unsupervised way, without the transcriptions of the training data or a pronunciation lexicon (Despotovic et al., 2015). AUDs are determined using a three-step approach: segmentation of the input speech into variable-length chunks of typically a few tens of milliseconds length; clustering the similar segments and assigning labels to clusters (AUDs); and iterative HMM training of obtained AUDs. Before the segmentation is carried out the Mel Frequency Cepstral Coefficient (MFCC) feature vectors are extracted from the raw speech, and the log energy and the first and second-order derivatives are appended to arrive at a 39-dimensional feature vector. Per utterance
ceptral mean and variance normalization is carried out.

The segmentation of the input speech into chunks is realized using a cosine distance as a local distance measure. A segment boundary is introduced if the value of the local distance measure between the mean representative of the current segment and the next feature vector is greater than a threshold. To prevent creating too short segments, the segments are constrained to the minimum length.

In the second step similar segments are clustered according to acoustic consistency using the unsupervised graph clustering algorithm by Newman and Girvan (2004). A label is assigned to each cluster to obtain an initial transcription of the spoken input in terms of sequences of cluster labels. These cluster labels will be denoted in further text as acoustic unit descriptors (AUDs).

The final step includes the iterative training of hidden Markov models (HMMs) for the discovered AUDs. Each AUD is modeled by a 3-state left-to-right hidden Markov model (HMM) with Gaussian mixture output densities. A zero gram language model is used to connect the AUDs.

It has been shown in Walter et al. (2014) that AUDs are able to capture acoustically consistent phenomena and represent recurring patterns of feature vectors, and furthermore that they are competitive to other unsupervised acoustic learning techniques. For more details about the learning of the acoustic representation the reader is referred to Walter et al. (2013).

### 2.2 Feature set

After obtaining an acoustic representation of the spoken utterance in terms of AUD sequences, we need to map it to a vector of fixed dimension, to be applicable to the machine learning methods used in this paper. The length of this vector is equal to the total number of different AUDs obtained for each particular speaker and it is created by setting each vector element to the occurrence frequency of the corresponding AUD. Binary features indicating only the presence of an AUD, disregarding its count, can also be used since early experiments have shown only a minor improvement due to including counts of AUDs. Moreover, the AUD feature set is augmented with AUD bigram counts. Higher n-grams were not included since they did not improve the classification performance and caused drastic increase in feature vector size.

This is basically a bag-of-n-grams approach, which ignores the relative position of the token (AUD unigram or bigram) in the utterance. Despite its simplicity, classification methods that use bag-of-n-grams or bag-of-words features often achieve high performance using state-of-the-art learning methods (Boulis and Ostendorf, 2005). Although more advanced features might lead to better classification performance, optimizing the feature extraction is beyond the scope of this paper. Our aim is to give a fair comparison of different machine learning approaches under the same or at least similar conditions.

### 3. Machine learning techniques for semantic analysis

In this section we give a brief outline for each of the machine learning algorithms used in this paper, with details of the particular implementation. While the well-known techniques such as e.g. MNB, SVM or LDA are given in less details, some task specific aspects of NMF and MLN are discussed more extensively.

#### 3.1. Multinomial naive Bayes

Multinomial naive Bayes is a special case of a naive Bayes classifier that is widely used in text classification. Whereas simple naive Bayes represents a data instance (spoken utterance) as the presence or absence of tokens (AUDs in our case), in MNB the data instance is represented by the number of token occurrences. The method is known from statistical language modeling for speech recognition as a unigram language model (McCallum and Nigam, 1998).

In semantic analysis the goal of the classifier is to find the best meaning representation for the spoken utterance represented by its attribute values. MNB assumes that the attribute values (AUDs) are independent of each other given the class (Manning et al., 2008). While the class-conditional independence assumption between predictors is obviously not true, it greatly simplifies the training process and often works very well in practice.

In a prediction step, the method computes the posterior probability of the unseen test data belonging to each class, and then assigns the observation to the class with the largest posterior probability.

#### 3.2. Support vector machines

Support vector machines attempt to do a binary classification by finding a decision boundary that is maximally far away from any data point between a linearly separable set of data. The decision boundary is the hyperplane defined as the linear decision function with maximal margin between data points belonging to different classes (Cortes and Vapnik, 1995). The support vectors represent a small subset of data points that lie on this margin; therefore they fully define the position of the hyperplane.

If the dataset is not linearly separable, we can map training vectors \( d \) into a higher dimensional space using the transform \( \phi(d) \), where the separation might be easier. We introduce the kernel function related to the transform \( \phi(d) \) with the relation \( k(d_i, d_j) = \phi(d_i)\phi(d_j) \). Hence, the decision boundary may be nonlinear in the original input space, but be a hyperplane in the transformed high-dimensional feature space. Some common kernel functions are linear, polynomial, radial basis function, sigmoid etc.

In case there are more than two classes, multiclass classification is implemented using one-against-one approach, where one SVM is constructed for each pair of classes (Milgram et al., 2006). Thus, \( c(c-1)/2 \) classifiers are constructed for \( c \) classes. Classification is performed according to the maximum voting criterion; the unknown entry is assigned to the class with the highest number of votes. We used the LIBSVM library for SVM implementation\(^1\) with a linear kernel (Chang and Lin, 2011).

#### 3.3. Maximum entropy

The maximum entropy method searches for a conditional probability distribution of the class label \( c \) given a data instance (utterance) \( d \) that is as uniform as possible under given constraints. Without any constraints the probability distribution would simply be uniform. Each constraint will move the distribution further away from being uniform, but closer to the data. Constraints on the conditional distribution are set from the training dataset using features. Let us define a feature \( f(d, c) \) as a real-valued function of the training data instance \( d \) and the class label \( c \). Similar as in MNB we can use token (AUD or AUD bigram) counts as features, where \( f_i \) is a function that equals zero if the token \( t \) does not appear in the utterance \( d \) and equal to the number of token occurrences \( N(d, t) \) otherwise:

\[
f(d, c) = \begin{cases} N(d, t), & \text{if } t \in d \\ 0, & \text{otherwise} \end{cases}
\]

In general, it is expected that in text classification problems features accounting for token occurrences are more beneficial compared to simple binary features (Nigam et al., 1999). The conditional distribution that an input data instance \( d \) belongs to a class \( c \) is defined as (Berger et al., 1996)

---

\(^1\) Available from http://www.csie.ntu.edu.tw/~cjlin/libsvm.
P(q|d) = \frac{1}{Z(d)} \exp \left( \sum_{i=1}^{N} \omega_{i}f_{i}(d, c) \right)

(2)

where \( \omega_{i} \) is a weight to be estimated and

\[ Z(d) = \sum_{c} \exp \left( \sum_{i=1}^{N} \omega_{i}f_{i}(d, c) \right) \]

(3)

is the normalizing term. Given this model, we wish to choose \( \omega_{i} \) that maximizes the conditional distribution of the data. The limited-memory quasi-Newton optimization algorithm (L-BFGS) was used to determine the optimal weights. In our work we used the MaxEnt implementation\(^2\) given in Weinman et al. (2011).

### 3.4. Linear discriminant analysis

Linear discriminant analysis is based upon the concept of searching for a linear combination of predictors that best separates between classes. Let us assume a set of training data instances \( d \in \mathbb{R}^{n} \) and class labels \( c \in \{c_{1}, c_{2}, ..., c_{K}\} \). Suppose we model each class conditional density as multivariate Gaussian, where each class \( c_{i} \) has its own mean \( \mu_{i} \), but shares a common covariance matrix \( \Sigma \). The means and the covariance matrix are estimated from the training data (Porter and Narsky, 2013).

To predict the classes of unseen test data instances the trained classifier finds the class by solving \( \hat{y} = \arg \max_{c} \{P(c|d)\} \), where \( P(c|d) \) is the posterior probability that a data instance \( d \) belongs to a class \( c_{i} \). Applying the Bayes rule we get \( \hat{y} = \arg \max_{c} \{P(d|c)P(c)\} \), where \( P(c) \) is the prior probability of class \( c_{i} \) and we assume that \( P(d|c_{i}) \) follows a multivariate Gaussian distribution. The decision boundary between classes \( c_{i} \) and \( c_{j} \) is defined with

\[ \delta_{ij}(d) = d^T\Sigma^{-1}(\mu_{i} - \mu_{j}) + c = 0 \]

(4)

where \( c = \log \frac{P(d)}{P(d)} - \frac{1}{2} (\mu_{i}^T\Sigma^{-1}\mu_{i} - \mu_{j}^T\Sigma^{-1}\mu_{j}) \) and \( \delta_{ij}(d) \) is a linear function with respect to \( d \) known as linear discriminant function, hence the name LDA.

### 3.5. Non-negative matrix factorization

Given a set of multivariate \( N \)-dimensional data vectors, non-negative matrix factorization decomposes an \( N \times M \) matrix \( V \), where \( M \) is the number of examples in the dataset (spoken utterances), into lower rank matrices \( W \) of size \( \frac{N}{K} \times R \) and \( H \) of size \( R \times M \), where typically \( R \ll M \) and \( N \ll R \), with the constraint that all three matrices are non-negative. In other words, each data vector in matrix \( V \) can be approximated by a linear combination of the columns of \( W \), weighted by the components of \( H \) (Lee and Seung, 2000).

\[ V \approx WH \]

(5)

Solution to Eq. (5) can be found by minimizing a cost function between \( V \) and \( WH \) using the Kullback–Leibler divergence as a distance measure (Ons et al., 2013a):

\[ D_{KL}(V||WH) = \sum_{i} \sum_{j} \left( v_{ij} \log \frac{v_{ij}}{(WH)_{ij}} - v_{ij} + (WH)_{ij} \right) \]

(6)

Convergence towards a local optimum is guaranteed using multiplicative update rules, as in Lee and Seung (1999).

When NMF is used in ASR, matrix \( W \) represents a dictionary matrix containing recurrent acoustic patterns (word-like units) and \( H \) is a matrix of activations of these patterns. Utterance-based fixed length feature vectors are required for NMF, which we obtain by transforming the acoustic feature vectors into AUD sequences and computing the histogram of occurrences of AUDs (Walter et al., 2014).

In addition to the acoustic representation, a weak form of utterance-based supervision is employed using a label matrix \( V \), which defines a semantic representation of the utterance given in the form of semantic frames that consists of slot values. Matrix \( V \) has \( K \) rows, where \( K \) is the number of labels and \( V_{ij} = n \) if label \( i \) occurs \( n \) times in utterance \( j \). The supervision information links the discovered acoustic patterns to labels and also helps NMF to avoid local optima of the Kullback–Leibler divergence (Ons et al., 2013b). Renaming \( V \) and \( W \) to \( V_{s} \) and \( W_{a} \) respectively, where index \( s \) denotes acoustic representation of the input speech, Eq. (5) can be rewritten as (Ons et al., 2014)

\[ V_{s} \approx W_{a}H \]

(7)

where \( W_{a}[K \times R] \) defines recurrent semantic patterns. During the training \( V_{s} \) is estimated from the training data, first \( K \times K \) entries in \( W_{a} \) are initialized as identity matrix, while last \( K \times (K - 1) \) entries are randomly initialized. First \( K \) rows in \( H \) are initialized with \( V_{s} \) and remaining \( K - K \) rows are randomly initialized. Solutions for \( W_{a} \) and \( H \) that minimize the distance measure are obtained using update formulas. Using \( W_{a} \) acquired in the training process and constructing \( V_{s} \) from the test dataset we use

\[ V_{s}^\text{test} \approx W_{a}^\text{test}H^\text{test} \]

(8)

to determine unknown matrix of labels activations

\[ H^\text{test} = \arg \min_{H^\text{test}} D_{KL}(V_{s}^\text{test}||W_{a}^\text{test}H^\text{test}) \]

(9)

Finally we are able to predict labels in the test dataset and reveal semantic representation of the unseen test utterances by determining the test label matrix using

\[ V_{s}^\text{test} \approx W_{a}^\text{test}H^\text{test} \]

(10)

### 3.6. Conditional random fields

Conditional random fields belong to a class of discriminative undirected probabilistic graphical models. In probabilistic graphical models the underlying probability distribution is represented in a graphical form, with a node for each random variable and an edge between two random variables. The absence of an edge indicates conditional independence between these variables. Although the graph structure can in general be arbitrary, the most common structure for sequential data is the first-order chain (Wallach, 2004).

CRFs are an extension to the MaxEnt model for sequential data. While MaxEnt assumes that observations can be structured (e.g., sequence of words), labels need to be atomic. In CRFs both observations and labels can be structured. Hence, CRF can take context into account; e.g., the linear chain CRF can predict sequences of labels for sequences of input data instances. In our case, since the semantic frame that represents meaning of the spoken utterance is often composed of several slots, the prediction of one slot value may depend on the choice of the previous slot value in the semantic frame.

The linear-chain CRF is a special case of CRF that obeys the Markov property between its neighbouring labels. The conditional distribution that an input data instance \( d \) belongs to a class \( c \) for linear-chain CRF can be defined as (Sutton and McCallum, 2011)

\[ P(c|d) = \frac{1}{Z(d)} \exp \left( \sum_{j=1}^{K} \sum_{k=1}^{N} \omega_{jk}f_{jk}(c_{j}, c_{j-1}, d_{j}) \right) \]

(11)

where \( c = c_{0}^{N} \) and \( d = d_{1}^{N} \) are label sequences and observation sequences respectively, \( f_{jk} \) and \( \omega_{jk} \) are feature functions and corresponding weight parameters, respectively, and

\(^2\)Available from http://www.cs.grinnell.edu/~weinman/code/
\[ Z(d) = \frac{1}{Z} \exp \left( \sum_{j=1}^{N} \sum_{k=1}^{K} \omega_{jk} f_j(c_j, c_{j-1}, d_j) \right) \]

is a normalizing term that sums over all label sequences. Index \( j \) in Eq. (11) specifies the position in the input sequence \( d \), indicating that each feature function can depend on observations from any time step. This makes CRFs naturally suited to exploit the dependencies between observations, such as neighbouring words in a sentence. Note that the weights \( \omega_{jk} \) are not dependent on the position \( j \). Moving the sum over the position sequences in front of the exponential function, we can see the direct connection to the factor graph representation in undirected graphical models:

\[ P(\zeta | d) = \frac{1}{Z(d)} \prod_{j=1}^{N} \Psi(d, c) \]

where each clique in the graph can be represented by a factor node with the factor (potential function)

\[ \Psi(d, c) = \exp \left( \sum_{k=1}^{K} \omega_{jk} f_j(c_j, c_{j-1}, d_j) \right) \]

The weight parameters \( \omega_{jk} \) must be estimated from the training data. We used the scaled conjugate gradient for learning the weight parameters, while inference was done using the forward-backward algorithm.

3.7. Markov logic networks

A Markov network (Markov random field) is a model for the joint distribution of a set of random variables \( D = (D_1, D_2, ..., D_N) \) (Taskar et al., 2007)

\[ P(d) = \frac{1}{Z} \exp \left( \sum_{i=1}^{F} \omega_i \sum_{g \in G} g(d) \right) = \frac{1}{Z} \exp \left( \sum_{i=1}^{F} \omega_i n_i(d) \right) \]

where \( F \) is the total number of FOL formulae, \( \omega_i \) are weights, \( G_i \) are groundings of the \( i \)-th FOL formula, \( g(d) \) is a binary function that takes value 1 if \( G_i \) is true and 0 otherwise. Hence, \( n_i(d) = \sum_{g \in G} g(d) \) simply counts the true groundings of \( i \)-th FOL formula.

The potentials are usually represented as a log-linear combination of a set of features \( \Psi = \exp \left( \sum_{j=1}^{F} (\omega f_j(d)) \right) \), hence Eq. (15) can be rewritten as

\[ P(d) = \frac{1}{Z} \exp \left( \sum_{j=1}^{N} \omega f_j(d) \right) \]

where \( \omega_j \) and \( f_j \) are weight parameters and feature functions respectively.

We consider in this paper a first-order logic extension of Markov networks called Markov logic networks. First-order logic (FOL) formulae are used to define the relations between variables and interpret semantics in a particular domain of interest. Let us for example consider a voice controlled home automation domain. The utterance command **Turn on the light** can be interpreted in FOL using a predicate **Turn** ("< device >", "< state >"), with the following assignment to variables: 

- `< device >` := light
- `< state >` := on.

The assignment of constants to variables is called grounding and the resulting ground predicate in this case is **Turn(light, on)**. The command **Turn off the TV** will therefore associate a different grounding to the same predicate **Turn(TV,off)**. Let us furthermore define a predicate that indicates presence of a particular keyword in an uttered command **HasWord** (< word >, < utterance >) with a different grounding for each constituent word within the particular utterance. If the uttered command contains a given word, then the predicate **HasWord** is true for the pair (< word >, < utterance >); otherwise it is false. We wish to infer a meaning or a semantic representation of the spoken command. Therefore, we define a FOL formula that associates a spoken utterance with a possible meaning:

\[ \text{HasWord(<word >, <utterance >) } \Rightarrow \text{Turn(<device >, <state >) } \]

FOL can be considered as a language to construct templates for undirected graphical models (Markov networks) (Richardson and Domingos, 2006). The network nodes in this architecture are ground predicates, and the edges are the logical connectives used to construct the formula. Thus, an MLN becomes a Markov network only with respect to a specific grounding. A potential function is associated to each formula, and takes value 1 when the formula is true or 0 when it is false. A weight is assigned to each grounding of the FOL formula in MLN, which is related to a probability that the formula is satisfied for a particular truth value assignments to all ground predicates. Hence, MLN can be defined as a set of weighted FOL formulas. The joint probability distribution over a set of random variables that correspond to the groundings of the predicates in FOL formulae is given as

\[ Z = \sum_{d \in D} \exp \left( \sum_{g \in G} g(d) \right) \]

where \( F \) is the total number of FOL formulae, \( \omega_i \) are weights, \( G_i \) are groundings of the \( i \)-th FOL formula, \( g(d) \) is a binary function that takes value 1 if \( G_i \) is true and 0 otherwise. Hence, \( n_i(d) = \sum_{g \in G} g(d) \) simply counts the true groundings of \( i \)-th FOL formula.

4. Experiments

For our experiments, we used task-oriented conversational data from the DOMOTICA 3 home automation domain and the PATCOR card game domain, collected in the framework of the ALADIN project (Gemmeke et al., 2019). For both domains participants were not restricted to any particular words or grammar during the training phase, but could express their commands freely. This allowed different expressions for the same command.

4.1. DOMOTICA 3

The DOMOTICA 3 speech corpus contains recordings of speakers with dysarthria controlling a home automation system. Participants were 5 male and 4 female, aged between 11 and 61 years, with an average age of 35, suffering from spastic quadriparesis, ataxic dysarthria, severe nasal dysarthria or multiple sclerosis. For all adult speakers, speech intelligibility scores were obtained by analysing the recorded speech using the automated tool (Midday, 2012). For one, child participant, speech intelligibility test was not conducted. A speech intelligibility score above 85 was considered as non-impaired, while a score below 70 was considered as severely impaired. All except two speakers were considered to utter dysarthric speech, two of them with severe dysarthria (Gemmeke et al., 2013).

The language of the corpus is Belgian Dutch. The corpus was collected in a Wizard-of-Oz study, where the subjects were asked to command 26 distinct actions for the home automation system, which was simulated in a 3D computer animation to ensure an unbiased...
choice of words and grammar by the user (Tessema et al., 2013). The total length of the dataset used in our experiments is approximately 4 hours of speech, with 2055 utterances spoken by 9 speakers, 228 per speaker on average.

A typical command in DOMOTICA 3 is: *ALADIN lichten in de woonkamer en keuken uit (ALADIN turn off the lights in the living room and kitchen)*. While the commands are fairly short, the major challenge of the dataset is the fact that pronunciation of dysarthric speakers deviates from the non-impaired ones: rate of speech is lower, segments are pronounced differently, pronunciation is less consistent (Sanders et al., 2002).

### 4.2. PATCOR

The PATCOR speech corpus contains recordings of non-pathological, normal speaking subjects playing a vocally guided card game patience (solitaire). Participants were 4 male and 4 female, aged between 23 and 73 years, with an average age of 37 (Gemmeke et al., 2013). The language of the corpus is Belgian Dutch. The average number of moves per game session is 55 (Tessema et al., 2013). The total length of the dataset is approximately 3 hours and 20 minutes, with 1912 utterances spoken by eight speakers, 239 per speaker on average.

A typical command in PATCOR is: *De harten boer op de klaveren dame (Put the Jack of hearts on the Queen of clubs)*. Note the importance of the order of words here, where the change of word order would change the meaning of the utterance. Note also that commands such as: *De zwarte dame naar de rode heer (Put the black Queen on the red King)* are present in the dataset, where clearly ambiguous mappings are possible for both the black Queen (spades or clubs) and the red King (hearts or diamonds). An additional challenge is the use of synonyms (e.g. Koning and Heer may refer to the same card).

### 4.3. Semantic representation

The training utterances are not mapped to semantic representations at the word-level, since this would be an expensive and time-consuming task. Only the action labels indicating the command that is performed are assigned to each utterance, without the need for literal transcriptions. Semantic frames are used to discover the meaning of the spoken utterances. A semantic frame is a data structure that is composed of slot and slot values, which are associated with the action that is expressed in the spoken command.

The semantic frame structures for DOMOTICA 3 and PATCOR datasets are shown in Tables 1 and 2 respectively (Tessema et al., 2013). Note that for the PATCOR dataset slots *FromSuit*, *FromValue*, *TargetSuit* and *TargetValue* define a chosen card, where 1 denotes an Ace and 13 denotes a King, *FromFoundation* and *TargetFoundation* define actions for the foundation stacks at the top, *FromHand* defines a pile of remaining cards used to get more cards if the player runs out of moves and *FromColumn* and *TargetColumn* denote actions for seven columns in the centre of the playing field. Also note that this is a general frame structure designed to represent a meaning of every possible command that might be spoken by any user in a particular domain and that for most of the users not all the slots will be necessary (for most of them only the first 6 slots of the *MoveCard* frame are used). An example semantic frame representation for the utterance *Put the Jack of hearts on the Queen of clubs* is shown in Fig. 1. The system picks a frame *MoveCard* that represents the meaning conveyed in the utterance and fills its slots *FromSuit*, *FromValue*, *TargetSuit* and *TargetValue* accordingly with the slot values. Not all the slots of the particular frame need to be filled, as the meaning is completely represented using only four slots.

### 4.4. Evaluation

Since the dysarthric speech is not consistent and has significant variation between speakers, speaker dependent training is applied for each user. Moreover, only limited amount of training data is available due to an increased effort and quick fatigue of the dysarthric speakers; hence a cross validation procedure should be used to assess the trained models. A five-fold cross-validation procedure was used in this paper, where the dataset is partitioned into five subsets, four of them being used for training and the remaining one for testing. The cross-validation procedure is repeated 5 times (folds), with each of the subsets used exactly once as the test dataset (Despotovic et al., 2015). The folds are created under the constraint that each slot value should occur at least once in each fold.Slot values that do not meet this constraint are excluded, meaning that the corresponding parts of spoken command are treated as filler words (Ons et al., 2013a).

As a performance measure we use the slot $F$-score, which is the harmonic mean of slot precision and slot recall. Slot $F$-score is a commonly used metric in semantic frame based SLU (Wang et al., 2011)

\[
F - \text{score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (19)
\]

Slot precision is the number of correctly detected slots divided by the total number of retrieved slots, while slot recall is defined as a number of correctly detected slots divided by the total number of slots

---

**Table 1**

<table>
<thead>
<tr>
<th>Frame</th>
<th>Slot</th>
<th>Slot-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open/Close</td>
<td>Action</td>
<td>open, close</td>
</tr>
<tr>
<td>On/Off</td>
<td>Action</td>
<td>on, off</td>
</tr>
<tr>
<td>TripleCommands</td>
<td>Object</td>
<td>1,2,3,...,9 **</td>
</tr>
<tr>
<td>IncreaseHeating</td>
<td>Range</td>
<td>1,2,3</td>
</tr>
</tbody>
</table>

* defines objects: bathroom door, bedroom door, front door, bedroom shutter, living-room door shutter, living-room window shutter. ** defines objects: bathroom light, bedroom light, living-room and kitchen light, kitchen light, kitchen stove light, kitchen table light, reading light, living-room light, all lights.

---

**Table 2**

<table>
<thead>
<tr>
<th>Frame</th>
<th>Slot</th>
<th>Slot-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveCard</td>
<td>FromSuit</td>
<td>spades, diamonds, hearts, clubs</td>
</tr>
<tr>
<td></td>
<td>FromValue</td>
<td>1,2,3,...,13</td>
</tr>
<tr>
<td></td>
<td>TargetSuit</td>
<td>spades, diamonds, hearts, clubs</td>
</tr>
<tr>
<td></td>
<td>TargetValue</td>
<td>1,2,3,...,13</td>
</tr>
<tr>
<td></td>
<td>FromFoundation</td>
<td>1,2,3,4</td>
</tr>
<tr>
<td></td>
<td>TargetFoundation</td>
<td>1,2,3,4</td>
</tr>
<tr>
<td></td>
<td>FromColumn</td>
<td>1,2,3,4,5,6,7</td>
</tr>
<tr>
<td></td>
<td>TargetColumn</td>
<td>1,2,3,4,5,6,7</td>
</tr>
<tr>
<td></td>
<td>FromHand</td>
<td>()</td>
</tr>
</tbody>
</table>

**Fig. 1.** The semantic representation for the utterance *Put the Jack of hearts on the Queen of clubs* shown as a tree representation.
in the reference semantic frame.

\[
\text{precision} = \frac{\text{correct slots}}{\text{retrieved slots}}
\]

\[
\text{recall} = \frac{\text{correct slots}}{\text{reference slots}}
\]

That means that only correctly filled slots are accounted; incorrectly filled slots and false empty slots are penalized (Ons et al., 2013a). Note that the resulting slot F-score is averaged over all folds.

5. Results and discussion

In most of the voice controlled applications the user is expected to speak a command from a predefined vocabulary and grammar, and the language model is defined in the form of domain-specific context-free grammar and the acoustic model is trained using the transcribed speech databases. Such models are not suitable for speakers with dysarthria, where pronunciation deviates from the standard one. Hence, an unsupervised approach is employed in this paper for learning the subword units (AUDs) from speech, without the need for a custom pronunciation lexicon or transcribed speech. Word segmentation step is skipped and the commands are learned directly from the recognized subword unit sequences, since the word boundaries are hard to distinguish in dysarthric speech. For each speaker we learn speaker dependent acoustic models of the AUDs from the raw speech, as described in Section 2.1. The number of AUDs per speaker varied between 77 and 113 AUDs for DOMOTICA 3 and 67 and 98 AUDs for the PATCOR dataset, depending on the outcome of the unsupervised clustering algorithm. Each spoken utterance is represented using the discovered sequence of AUDs (Walter et al., 2014). After tokenization (tokens are subword unit sequences, since the word boundaries are hard to distinguish and speech intelligibility test was not conducted for child participants. ** The highest F-score for each speaker is highlighted.

![Image](image_url)

<table>
<thead>
<tr>
<th>Speaker</th>
<th>17</th>
<th>28</th>
<th>29</th>
<th>30</th>
<th>31</th>
<th>34</th>
<th>35</th>
<th>41</th>
<th>44</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender</td>
<td>F</td>
<td>F</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>F</td>
<td>F</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>Intelligibility</td>
<td>88.6</td>
<td>73.1</td>
<td>73.6</td>
<td>69</td>
<td>69</td>
<td>72.3</td>
<td>64.2</td>
<td>89.2</td>
<td>75.8</td>
<td></td>
</tr>
<tr>
<td># Utterances</td>
<td>347</td>
<td>204</td>
<td>174</td>
<td>198</td>
<td>198</td>
<td>225</td>
<td>144</td>
<td>164</td>
<td>228</td>
<td></td>
</tr>
<tr>
<td>MBN</td>
<td>90.2</td>
<td>75</td>
<td>86.5</td>
<td>84.4</td>
<td>67.2</td>
<td>88.8</td>
<td>93.6</td>
<td>80.3</td>
<td>93.1</td>
<td>84.3</td>
</tr>
<tr>
<td>SVM</td>
<td>95.4</td>
<td>74.5</td>
<td>89.7</td>
<td>85.7</td>
<td>69.2</td>
<td>91.2</td>
<td>95.5</td>
<td>77.9</td>
<td>91.9</td>
<td>85.7</td>
</tr>
<tr>
<td>MaxEnt</td>
<td>94.2</td>
<td>76.7</td>
<td>93.6</td>
<td>84.2</td>
<td>70.5</td>
<td>89</td>
<td>94.4</td>
<td>81.6</td>
<td>95.7</td>
<td>86.6</td>
</tr>
<tr>
<td>F-score</td>
<td>90.3</td>
<td>76.3</td>
<td>88.3</td>
<td>87.8*</td>
<td>67.9</td>
<td>93.1</td>
<td>92</td>
<td>78.6</td>
<td>95.1</td>
<td>85.5</td>
</tr>
<tr>
<td>LDA</td>
<td>96</td>
<td>76.2</td>
<td>94.9</td>
<td>84.4</td>
<td>72.7</td>
<td>87.8</td>
<td>95.4</td>
<td>84.5</td>
<td>91.7</td>
<td>87.1</td>
</tr>
<tr>
<td>NMF</td>
<td>96</td>
<td>76.3</td>
<td>93.8</td>
<td>87.1</td>
<td>72.1</td>
<td>93</td>
<td>96.3</td>
<td>82.4</td>
<td>96.1</td>
<td>88.2</td>
</tr>
<tr>
<td>CRF</td>
<td>96</td>
<td>76.7</td>
<td>93.8</td>
<td>87.1</td>
<td>72.1</td>
<td>93</td>
<td>96.3</td>
<td>82.4</td>
<td>96.1</td>
<td>88.2</td>
</tr>
<tr>
<td>MLN</td>
<td>98</td>
<td>83.9</td>
<td>97.3</td>
<td>86.7</td>
<td>75.7</td>
<td>96.6</td>
<td>97.7</td>
<td>86.2</td>
<td>99</td>
<td>91.2</td>
</tr>
</tbody>
</table>

*Speech intelligibility test was not conducted for child participants. ** The highest F-score for each speaker is highlighted.
The highest F-score for each speaker is highlighted.

Fig. 2. The results are obtained using the best performing machine learning algorithm for speaker 30 (LDA) using a combination of unigram and bigram features. Precision and recall were averaged over all folds. While precision measures the relevance of the inferred slot values, recall assesses how many correctly inferred slot values were returned. A good classifier returns both accurately retrieved slot values (high precision), as well as accurately returns a majority of all slot values (high recall), which is the case for slot values ‘On’ (Aan), ‘Bathroom light’ (Badkamer licht), ‘Headrest’ (Hoofdeinde), ‘Living-room door shutter’ (Rolluik woonkamer deur), ‘Bedroom light’ (Slaapkamer licht), ‘Standing lamp’ (Staande lamp), ‘Off’ (Uit) and ‘Increase heating’ (Verwarming hoger). A classifier with high precision and low recall retrieves only few results, but most of them accurately, as in ‘Reading lamp’ (Leeslamp) or ‘Bedroom door’ (Slaapkamerdeur) slot values. A classifier with high recall and low precision retrieves most of the results, but some of them incorrectly, as in ‘Living-room light’ (Woonkamer licht) slot value. Analyzing the confusion matrix we found that the misclassification often occurs for commands that are pronounced similarly, such as e.g. ‘Stand1’ and ‘Stand2’, where false negatives in case of one slot value (leading to low recall) cause false positives in case of the other one (leading to low precision).

Since MLNs have proved to have either superior (for speakers with dysarthria) or of comparable performance (for normal speaking subjects), we employ further improvements only to MLNs. Let us denote an MLN model that we presented here a compositional MLN, since frames are composed of slots, which are composed of slot values. Analysing the confusion matrix we noted that a significant source of error in compositional MLN was the inference of unwanted additional slots that

Table 4
Intelligibility scores and slot F-scores for DOMOTICA 3 dataset using unigram and bigram features.

<table>
<thead>
<tr>
<th>Speaker</th>
<th>17</th>
<th>28</th>
<th>29</th>
<th>30</th>
<th>31</th>
<th>34</th>
<th>35</th>
<th>41</th>
<th>44</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender</td>
<td>F</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>F</td>
<td>M</td>
<td>M</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td># Utterances</td>
<td>274</td>
<td>769</td>
<td>260</td>
<td>278</td>
<td>221</td>
<td>247</td>
<td>223</td>
<td>240</td>
<td>239</td>
<td></td>
</tr>
<tr>
<td>MNB</td>
<td>58.4</td>
<td>77.4</td>
<td>73.8</td>
<td>58.5</td>
<td>79.9</td>
<td>54.6</td>
<td>62.3</td>
<td>47.9</td>
<td>64.1</td>
<td></td>
</tr>
<tr>
<td>SVM</td>
<td>56.8</td>
<td>79.8</td>
<td>76.1</td>
<td>54.6</td>
<td>84.6</td>
<td>45.1</td>
<td>62.4</td>
<td>43.2</td>
<td>62.8</td>
<td></td>
</tr>
<tr>
<td>F-score</td>
<td>MaxEnt</td>
<td>60.6</td>
<td>78.1</td>
<td>79.7</td>
<td>53.9</td>
<td>87.2</td>
<td>54.2</td>
<td>71.3</td>
<td>49.3</td>
<td>66.8</td>
</tr>
<tr>
<td>LDA</td>
<td>61.9</td>
<td>78.4</td>
<td>77.5</td>
<td>58.1</td>
<td>90.8</td>
<td>55.9</td>
<td>68.3</td>
<td>48.4</td>
<td>67.4</td>
<td></td>
</tr>
<tr>
<td>NMF</td>
<td>61.9</td>
<td>66.7</td>
<td>74.6</td>
<td>50.1</td>
<td>89.4</td>
<td>46.7</td>
<td>69.6</td>
<td>41.3</td>
<td>62.5</td>
<td></td>
</tr>
<tr>
<td>CRF</td>
<td>63.3</td>
<td>78.2</td>
<td>81.8</td>
<td>59.7</td>
<td>87.1</td>
<td>56.2</td>
<td>71</td>
<td>52.2</td>
<td>68.7</td>
<td></td>
</tr>
<tr>
<td>MLN</td>
<td>62.5</td>
<td>77.1</td>
<td>79.6</td>
<td>62.9</td>
<td>88.9</td>
<td>45.8</td>
<td>73</td>
<td>51.1</td>
<td>67.6</td>
<td></td>
</tr>
</tbody>
</table>

Table 5
Slot F-scores for PATCOR dataset using unigram features.

<table>
<thead>
<tr>
<th>Speaker</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender</td>
<td>F</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>F</td>
<td>F</td>
<td>M</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td># Utterances</td>
<td>274</td>
<td>769</td>
<td>260</td>
<td>278</td>
<td>221</td>
<td>247</td>
<td>223</td>
<td>240</td>
<td>239</td>
</tr>
<tr>
<td>MNB</td>
<td>58.4</td>
<td>77.4</td>
<td>73.8</td>
<td>58.5</td>
<td>79.9</td>
<td>54.6</td>
<td>62.3</td>
<td>47.9</td>
<td>64.1</td>
</tr>
<tr>
<td>SVM</td>
<td>56.8</td>
<td>79.8</td>
<td>76.1</td>
<td>54.6</td>
<td>84.6</td>
<td>45.1</td>
<td>62.4</td>
<td>43.2</td>
<td>62.8</td>
</tr>
<tr>
<td>F-score</td>
<td>MaxEnt</td>
<td>60.6</td>
<td>78.1</td>
<td>79.7</td>
<td>53.9</td>
<td>87.2</td>
<td>54.2</td>
<td>71.3</td>
<td>49.3</td>
</tr>
<tr>
<td>LDA</td>
<td>61.9</td>
<td>78.4</td>
<td>77.5</td>
<td>58.1</td>
<td>90.8</td>
<td>55.9</td>
<td>68.3</td>
<td>48.4</td>
<td>67.4</td>
</tr>
<tr>
<td>NMF</td>
<td>61.9</td>
<td>66.7</td>
<td>74.6</td>
<td>50.1</td>
<td>89.4</td>
<td>46.7</td>
<td>69.6</td>
<td>41.3</td>
<td>62.5</td>
</tr>
<tr>
<td>CRF</td>
<td>63.3</td>
<td>78.2</td>
<td>81.8</td>
<td>59.7</td>
<td>87.1</td>
<td>56.2</td>
<td>71</td>
<td>52.2</td>
<td>68.7</td>
</tr>
<tr>
<td>MLN</td>
<td>62.5</td>
<td>77.1</td>
<td>79.6</td>
<td>62.9</td>
<td>88.9</td>
<td>45.8</td>
<td>73</td>
<td>51.1</td>
<td>67.6</td>
</tr>
</tbody>
</table>

*Speech intelligibility test was not conducted for child participants. ** The highest F-score for each speaker is highlighted.
falsey remained above a predefined threshold. It turned out to be very hard to define a reasonable rejection threshold, which would be a good trade-off between inferred and rejected slots within a frame. Hence, we employed a different approach, where the hard threshold is avoided by mapping the spoken utterance to a null slot value for all the unwanted slots during the training phase. In this way we not only learn to which slots the utterance is mapped, we also learn to which ones it should not be mapped. This resulted in an additional absolute improvement in $F$-score of over 3% on average for the PATCOR dataset. This setup was not be mapped. This resulted in an additional absolute improvement in $F$-score for each speaker is highlighted.

The major drawback is the fact that standard learning algorithms for the task of semantic analysis of spoken input, with an emphasis on application in dysarthric speech, where the amount of training data is low. Probabilistic undirected graphical models, such as Markov logic networks and linear-chain conditional random fields, have shown to substantially outperform all other algorithms tested in this paper. Moreover, MLNs have proved to be extremely robust to recognition errors, which are caused by imperfect articulation in dysarthric speech. Coupled with an unsupervised learning of speech representations the approach is especially applicable for the semantic analysis in the presence of noisy and inconsistent input data.

The major drawback is the fact that standard learning algorithms for both CRFs and MLNs are very slow and do not scale well to large amounts of training data. The problem is partly addressed here for MLNs using a hierarchical approach, which decomposes a larger task into a series of smaller tasks where learning may be more tractable. These constituent models are considerably faster to train than a full MLN. However, the application to large and complex domains is still limited.

The results obtained in this study are encouraging as the unsupervised learning of subword units, accompanied with weakly supervised semantic analysis, where training utterances require only a semantic label, allows possibility of recording more data, avoiding the need for expensive literal transcriptions. However, the datasets considered in this study were of quite limited semantic variability. It remains a question for future research to investigate how well the proposed approach generalizes to semantically more variable tasks.

**Acknowledgements**

This work was partly funded by DFG, contract no. Ha 3455/9-1, within the Priority Program SPP1527 Autonomous Learning.

Vladimir Despotovic was supported by an Erasmus Mundus Action 2
scholarship within the EUROWEB scholarship programme.

We wish to thank Jort Gemmeke and Hugo Van hamme from KU Leuven for making available to us the datasets used in this paper.

References


Madison, WI, USA. pp. 41–48.


Milgram, J., Cheriet, M., Sabourin, R., 2006. “One against one” or “One against all” which is one better for handwriting recognition with SVMs? Tenth International Workshop on Frontiers in Handwriting Recognition, La Baule, France.

