Speech Communication 99 (2018) 242–251

Contents lists available at ScienceDirect

Speech Communication

journal homepage: www.elsevier.com/locate/specom

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

Vladimir Despotovic*,[a], Oliver Walter[b], Reinhold Haeb-Umbach[b]

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

ARTICLE INFO

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

ABSTRACT

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. interpunction). 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

* Corresponding author.
E-mail address: vdespotovic@tfbor.bg.ac.rs (V. Despotovic).

https://doi.org/10.1016/j.specom.2018.04.005
Received 3 April 2017; Received in revised form 8 March 2018; Accepted 15 April 2018
Available online 16 April 2018
0167-6393/

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 (Mairesse et al., 2009). Wutiwiwatchai 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 (Wutiwiwatchai and Furui, 2003). Ons, Gemmeke and Van hamme propose an NMF-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 (DOMOTICA 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 cepstral 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 zerogram 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_i into a higher dimensional space using the transform $\varphi(d_i)$, where the separation might be easier. We introduce the kernel function related to the transform $\varphi(d_i)$ with the relation $k(d_i, d_j) = \varphi(d_i)\varphi(d_j)$. 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 $f_i(d, c)$ 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:

$$f_i(d, c) = \begin{cases} N(d, t), & \text{if } t \in d \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

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 .

$$P(c|d) = \frac{1}{Z(d)} exp\left(\sum_{i=1}^{n} \omega_i f_i(d, c) \right) \quad (2)$$

where ω_i i is a weight to be estimated and

$$Z(d) = \sum_{c} exp\left(\sum_{i=1}^{n} \omega_i f_i(d, c) \right) \quad (3)$$

is the normalizing term. Given this model, we wish to choose ω_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 $d \in R^n$ and class labels $c \in \{c_1, c_2, \dots, c_K\}$. Suppose we model each class conditional density as multivariate Gaussian, where each class c_i has its own mean μ_i, but shares a common covariance matrix Σ. 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 $\hat{y} = \arg\max_i(P(c_i|d))$, where $P(c_i|d)$ is the posterior probability that a data instance d belongs to a class c_i. Applying the Bayes rule we get $\hat{y} = \arg\max_i((P(c_i)P(d|c_i))$, where $P(c_i)$ 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

$$\delta_{ij}(d) = d^T \Sigma^{-1}(\mu_i - \mu_j) + c = 0 \quad (4)$$

where $c = \log\frac{P(c_i)}{P(cj)} - \frac{1}{2}\left(\mu_i^T \Sigma^{-1}\mu_i - \mu_j^T \Sigma^{-1}\mu_j\right)$ 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 NxM matrix V, where M is the number of examples in the dataset (spoken utterances), into lower rank matrices W of size NxR and H of size RxM, where typically $R << N$ and $R << M$, 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)

$$V \approx W \cdot H \quad (5)$$

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

$$D_{KL}(V||WH) = \sum_{i}\sum_{j}\left(V_{ij} \log\frac{V_{ij}}{(WH)_{ij}} - V_{ij} + (WH)_{ij} \right) \quad (6)$$

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_s which defines a semantic representation of the utterance given in the form of semantic frames that consists of slot values. Matrix V_s has K rows, where K is the number of labels and $V_{s,ij} = n$ 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_a and W_a respectively, where index a denotes acoustic representation of the input speech, Eq. (5) can be rewritten as (Ons et al., 2014)

$$\begin{bmatrix} V_s \\ V_a \end{bmatrix} \approx \begin{bmatrix} W_s \\ W_a \end{bmatrix} \cdot H \quad (7)$$

where W_s [KxR] defines recurrent semantic patterns. During the training V_s and V_a are estimated from the training data, first KxK entries in W_s are initialized as identity matrix, while last Kx($R - K$) entries are randomly initialized. First K rows in H are initialized with V_s and remaining $R - K$ rows are randomly initialized. Solutions for W_s, 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_a from the test dataset we use

$$V_a^{\text{test}} \approx W_a \cdot H^{\text{test}} \quad (8)$$

to determine unknown matrix of labels activations

$$H^{\text{test}} = \underset{H^{\text{test}}}{\arg\min}\ D_{KL}(V_a^{\text{test}}||W_a H^{\text{test}}) \quad (9)$$

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

$$V_s^{\text{test}} \approx W_s \cdot H^{\text{test}} \quad (10)$$

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)

$$P(c|d) = \frac{1}{Z(d)} exp\left(\sum_{j=1}^{N}\sum_{k=1}^{K} \omega_k f_k(c_j, c_{j-1}, d_j) \right) \quad (11)$$

where $c = c_j|_{j=1}^N$ and $d = d_j|_{j=1}^N$ are label sequences and observation sequences respectively, $f_k|_{k=1}^K$ and $\omega_k|_{k=1}^K$ are feature functions and corresponding weight parameters, respectively, and

[2] Available from http://www.cs.grinnell.edu/~weinman/code/ .

$$Z(d) = \sum_{c} exp\left(\sum_{j=1}^{N}\sum_{k=1}^{K} \omega_k f_k(c_j, c_{j-1}, d_j)\right) \tag{12}$$

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 ω_k are not dependent on the position j. Moving the sum over the sequence positions in front of the exponential function, we can see the direct connection to the factor graph representation in undirected graphical models

$$P(c|d) = \frac{1}{Z(d)} \prod_{j=1}^{N} \Psi_j(d, c) \tag{13}$$

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

$$\Psi_j(d, c) = exp\left(\sum_{k=1}^{K} \omega_k f_k(c_j, c_{j-1}, d_j)\right). \tag{14}$$

The weight parameters w_k 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)

$$P(d) = \frac{1}{Z} \prod_{j=1}^{N} \Psi_j(d) \tag{15}$$

where d is an assignment of values to D, Ψ_j is a potential function and $Z = \sum_{d\in D}\prod_{j=1}^{N}\Psi_j(d)$ is a normalization constant known as partition function. There is one node for each variable in the undirected graph and the model has a potential function for each clique in the graph. Each variable is conditionally independent of all other variables given its immediate neighbours. The potentials are usually represented as a log-linear combination of a set of features $\Psi_j = exp\left(\sum_{j=1}^{N}(\omega_j f_j(d))\right)$, hence Eq. (15) can be rewritten as

$$P(d) = \frac{1}{Z} exp\left(\sum_{j=1}^{N} \omega_j f_j(d)\right) \tag{16}$$

where ω_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 uttered command *Turn on the light* can be interpreted in FOL using a predicate *Turn* (< *device* > , < *state* >), with the following assignment to variables: < *device* > ≔ *light* and < *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

$$HasWord(<word>, <utterance>) \Rightarrow Turn(<device>, <state>) \tag{17}$$

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

$$P(d) = \frac{1}{Z} exp\left(\sum_{i=1}^{F} \omega_i \sum_{g\in G_i} g(d)\right) = \frac{1}{Z} exp\left(\sum_{i=1}^{F} \omega_i n_i(d)\right) \tag{18}$$

where F is the total number of FOL formulae, ω_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, $n_i(d) = \sum_{g\in G_i} g(d)$ simply counts the true groundings of i-th FOL formula. $Z = \sum_{d\in D} exp(\sum_{i=1}^{F}\omega_i n_i(d))$ is a normalizing term obtained by summing over all possible groundings of the predicates.

Weights ω_i are typically learned from training data. We used the Alchemy 2.0 engine[3] (Kok et al., 2009) for learning the weights discriminatively using the rescaled conjugate gradient algorithm, while the inference in MLN was performed using the MC-SAT algorithm (Poon and Domingos, 2006).

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., 2013). 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 (Middag, 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

[3] Available from https://alchemy.cs.washington.edu .

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-impared 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.

Table 1
DOMOTICA 3 semantic frame structure.

Frame	Slot	Slot-value
Open/Close	Action	open, close
	Object	1,2,3,... ,6 *
On/Off	Action	on, off
	Object	1,2,3,... ,9 **
TripleCommands	Object	headrest, standing lamp
	Range	1,2,3
IncreaseHeating	{ }	{ }

* 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
PATCOR semantic frame structure.

Frame	Slot	Slot-value
MoveCard	FromSuit	spades, diamonds, hearts, clubs
	FromValue	1,2,3,... ,13
	TargetSuit	spades, diamonds, hearts, clubs
	TargetValue	1,2,3,... ,13
	FromFoundation	1,2,3,4
	TargetFoundation	1,2,3,4
	FromColumn	1,2,3,4,5,6,7
	TargetColumn	1,2,3,4,5,6,7
	FromHand	{ }
DealCard	{ }	{ }

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)

$$F - score = 2\cdot\frac{precision\cdot recall}{precision + recall} \tag{19}$$

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

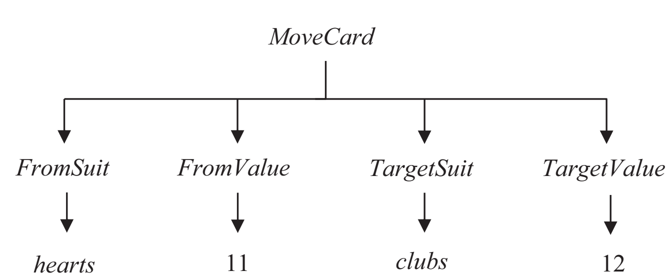

Fig. 1. The semantic representation for the utterance *Put the Jack of hearts on the Queen of clubs* shown as a tree representation.

in the reference semantic frame.

$$precission = \frac{correct\ slots}{retrieved\ slots} \tag{20}$$

$$recall = \frac{correct\ slots}{reference\ slots} \tag{21}$$

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, where 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 AUDs) we find unigram and combination of unigram and bigram features, which we use as input to the learning algorithms.

An additional challenge is imposed by the request that the users are allowed to express their command freely, without the need to use the exact, predefined words. The system is trained using the examples spoken by the user. An example of freely spoken command is:

"*The **heating** system should be **turn**ed **off**.*"

While some parts of the phrase are informative and can be directly mapped to commands (e.g. "heating" and "turn off"), there are also non-informative parts (e.g. "system should be") which should be ignored. The system needs to find the spoken keywords in the phrase and learn the association between the keyword and the label associated with the particular action. However, these mappings are noisy, as the commands in the training dataset are not always consistent and the phrases, such as e.g. *"Turn off, please!"* are also possible, where the keyword that denotes the preferred device is obviously missing. The association of spoken keywords to commands is a machine learning problem, where the command labels introduce a weak form of supervision to the machine learning process. For learning mappings to semantic representations we use seven machine learning techniques: multinomial naive Bayes, support vector machines, maximum entropy, linear discriminant analysis, non-negative matrix factorization, conditional random fields and Markov logic networks.

For MLN we need to define a set of first-order rules which map the AUD sequences to its semantic frame representation. We give an example for the PATCOR dataset, but employ a similar analogy to DOMOTICA 3 dataset.

$$\begin{aligned}
&HasAUD(+a, u) \ => \ FromSuit(+s, u)\\
&HasAUD(+a, u) \ => \ FromValue(+v, u)\\
&HasAUD(+a, u) \ => \ TargetSuit(+s, u)\\
&HasAUD(+a, u) \ => \ TargetValue(+v, u)\\
&HasAUD(+a, u) \ => \ FromFoundation(+f, u)\\
&HasAUD(+a, u) \ => \ TargetFoundation(+f, u)\\
&HasAUD(+a, u) \ => \ FromColumn(+c, u)\\
&HasAUD(+a, u) \ => \ TargetColumn(+c, u)\\
&HasAUD(+a, u) \ => \ FromHand(u)\\
&HasAUD(+a, u) \ => \ DealCard(u)
\end{aligned}$$

where $s \in \{spades, diamonds, hearts, clubs\}$, $v \in \{1, 2, ..., 13\}$, $f \in \{1, 2, 3, 4\}$ and $c \in \{1, 2, ..., 7\}$. The *HasAUD(a, u)* predicate is an evidence predicate, which states that a particular AUD (or AUD bigram) *a* is part of the AUD sequence (utterance) *u*. The predicates at the right side of the implication operator define the mapping of the AUD sequence *u* to a particular slot value. The + operator is a per constant operator that produces a separate clause for each combination of *a* and slot value. A separate weight is also learned for each clause obtained in this way. For more details on the MLN structure the reader is referred to Despotovic et al. (2015). Finally, we infer the probabilities of mapping the AUD sequence to each of the slot values given in Table 2. The slot value with the highest probability is chosen for every slot only if it is higher than a predefined threshold; hence not all the slots need to be inferred for a semantic frame.

Obtained slot *F*-scores for all the tested algorithms for DOMOTICA 3 dataset using unigram and a combination of unigram and bigram features are shown in Tables 3 and 4, respectively. The best scores for each speaker are highlighted. MLNs have a dominant performance for all but one speaker, outperforming CRFs as the second best technique for 3% on average using only unigram features. Adding bigrams improves the scores for all the algorithms in the range of 1.7% for NMF to 3.7% for LDA. MLNs and CRFs are again dominant over the rest of techniques, once more MLN having a 3% better *F*-score than CRF. Note that the best *F*-scores are obtained for speakers with the highest intelligibility scores (17 and 44), while severely impaired speakers with intelligibility scores below 70 (30 and 41) have lower scores, as expected. However, observing speakers 28, 29 and 35 with similar intelligibility scores, but a substantial difference in performance, leads to the conclusion that consistency in speech production might be as important as intelligibility in achieving good recognition scores (Doyle et al., 1997).

Precision and recall across different slot values for one speaker with severe dysarthria (speaker 30) in DOMOTICA 3 dataset is shown in

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

Speaker		17	28	29	30	31	34	35	41	44	Average
Gender		F	F	M	M	M	M	F	F	M	
Intelligibility		88.6	73.1	73.6	69	NA*	76.2	72.3	64.2	89.2	75.8
# Utterances		347	204	174	198	225	331	268	144	164	228
	MNB	90.2	75	86.5	84.4	67.2	88.8	93.6	80.3	93.1	84.3
	SVM	95.4	74.5	89.7	85.7	69.2	91.2	95.5	77.9	91.9	85.7
	MaxEnt	94.2	76.7	93.6	84.2	70.5	89	94.4	81.6	95.7	86.6
F-score	LDA	90.3	76.3	88.3	**87.8****	67.9	93.1	92	78.6	95.1	85.5
	NMF	96	76.2	94.9	84.4	72.7	87.8	95.4	84.5	91.7	87.1
	CRF	96.3	76.7	93.8	87.1	72.1	93	96.3	82.4	96.1	88.2
	MLN	**98**	**83.9**	**97.3**	86.7	**75.7**	**96.6**	**97.7**	**86.2**	**99**	**91.2**

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

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

Speaker		17	28	29	30	31	34	35	41	44	Average
Gender		F	F	M	M	M	M	F	F	M	
Intelligibility		88.6	73.1	73.6	69	NA*	76.2	72.3	64.2	89.2	75.8
# Utterances		347	204	174	198	225	331	268	144	164	228
	MNB	94.1	79.3	88.8	85.2	73.8	91.4	95.3	86.2	96	87.8
	SVM	97.6	76.6	93.5	86.1	73.3	93	97.4	85.2	95.8	88.7
F-score	MaxEnt	94.9	79.8	95.9	89.1	74.7	93.3	96.7	83.1	96.6	89.4
	LDA	93.2	80.4	90.3	**92.5****	76.1	93.7	95.9	86.3	94.6	89.2
	NMF	96.3	82	94.5	87.6	74.3	90.2	94.6	86.8	93.3	88.8
	CRF	97.7	81.1	96	90.1	76.5	94.3	97.3	87.6	**98.8**	91.1
	MLN	**98.5**	**90.6**	**97**	92.1	**83.1**	**96.2**	**98.8**	**91.6**	**98.8**	**94.1**

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

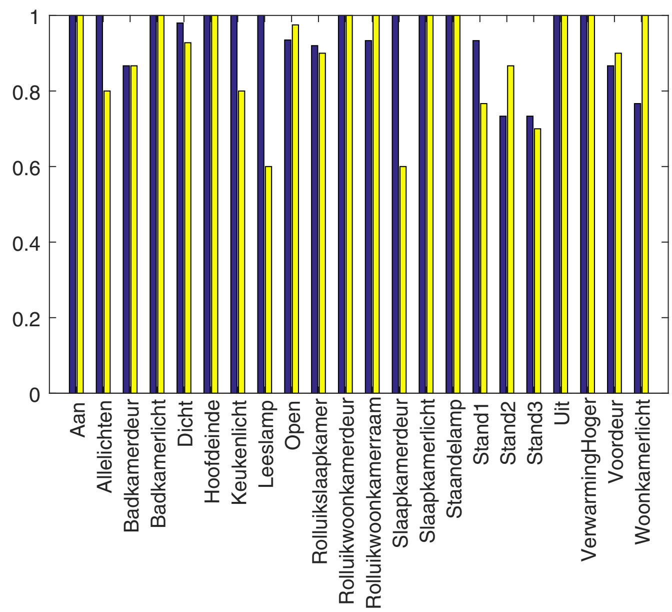

Fig. 2. Precision and recall (left bar - precision; right bar - recall) for different slot values for DOMOTICA 3 dataset (speaker 30).

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).

No explicit method was introduced to improve robustness to recognition errors, due to deviated pronunciation of speakers with dysarthria. However, it seems that robustness to noisy input data is an inherent property of some of the employed machine learning algorithms, especially MLNs. To understand the underlying robustness of MLNs, one needs to comprehend its learning process. A separate weight is learned for mapping of each AUD belonging to each training utterance, to a particular slot value. When the recognition error occurs during training, the weight associated to this mapping (FOL formula) is smaller, as there are fewer bad examples in the training dataset. Hence, the confidence that this FOL formula is true is lower compared to FOL formula of the error-free examples.

Results for the PATCOR dataset using unigram features show that the best *F*-score is obtained using CRFs, outperforming MLNs and LDA by approximately 1% (see Table 5). Adding the bigrams improves the performance for all the tested algorithms, leading to best scores obtained using CRFs and MLNs (see Table 6). Early experiments with higher-order *n*-grams did not further improve the performance; hence they were not taken into account.

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 inferred semantic frames we noted that a significant source of error in compositional MLN was the inference of unwanted additional slots that

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

Speaker		1	2	3	4	5	6	7	8	Average
Gender		F	M	M	F	F	M	M	F	
# Utterances		274	169	260	278	221	247	223	240	239
	MNB	58.4	77.4	73.8	58.5	79.9	54.6	62.3	47.9	64.1
	SVM	56.8	**79.8***	76.1	54.6	84.6	45.1	62.4	43.2	62.8
F-score	MaxEnt	60.6	78.1	79.7	53.9	87.2	54.2	71.3	49.3	66.8
	LDA	61.9	78.4	77.5	58.1	**90.8**	55.9	68.3	48.4	67.4
	NMF	61.9	66.7	74.6	50.1	89.4	46.7	69.6	41.3	62.5
	CRF	**63.3**	78.2	**81.8**	59.7	87.1	**56.2**	71	**52.2**	**68.7**
	MLN	62.5	77.1	79.6	**62.9**	88.9	45.8	**73**	51.1	67.6

*The highest *F*-score for each speaker is highlighted.

Table 6
Slot *F*-scores for PATCOR dataset using unigram and bigram features.

Speaker		1	2	3	4	5	6	7	8	Average
Gender		F	M	M	F	F	M	M	F	
# Utterances		274	169	260	278	221	247	223	240	239
F-score	MNB	61.8	81.6	74.7	62	86.5	56.7	72.7	48.3	68
	SVM	59.7	80	77.8	57.5	89.4	49.9	65.9	45.1	65.7
	MaxEnt	64.7	82.5	82.4	63.1	90.9	60.4	75.7	53.3	71.6
	LDA	67.7	82.9	77.7	57.8	**93.3**	59.9	75.2	51.6	70.8
	NMF	66.1	69.3	76.2	55.9	90.9	54.7	77	48.5	67.3
	CRF	**68.3**	**83.2**	**83.2**	62.8	89.9	**61.5**	73	55.4	**72.2**
	MLN	66.3	82.9	80.6	**65.1***	91.4	54.8	**79.1**	**56.5**	72.1

*The highest *F*-score for each speaker is highlighted.

Table 7
Slot *F*-scores for DOMOTICA 3 dataset using different MLN setups.

	Speaker	17	28	29	30	31	34	35	41	44	Average
Unigram	MLN - compositional	98	83.9	97.3	86.7	75.7	96.6	97.7	86.2	99	91.2
	MLN - hierarchical	97.7	83.2	97.3	88.3	75	96	96	87.9	99	91.1
Unigram	MLN - compositional	98.5	90.6	97	92.1	83.1	96.2	98.8	91.6	98.8	94.1
+Bigram	MLN - hierarchical	98.2	90	96.6	90	81.5	96.7	98.8	91.8	99	93.6

Table 8
Slot *F*-scores for PATCOR dataset using different MLN setups.

	Speaker	1	2	3	4	5	6	7	8	Average
Unigram	MLN - compositional	62.5	77.1	79.6	62.9	88.7	45.8	73	51.1	67.6
	MLN - compositional (*null*)*	64.1	79.7	81.6	65.2	93.2	55.5	76.1	52.2	70.9
	MLN - hierarchical (*null*)*	65.1	78.2	81.4	65	92.3	55.3	76.4	50.3	70.5
Unigram +Bigram	MLN - compositional	66.3	82.9	80.6	65.1	91.4	54.8	79.1	56.5	72.1
	MLN - compositional (*null*)*	68.4	85.6	83.6	67.4	94.5	65.1	81.4	56.1	75.3
	MLN - hierarchical (*null*)*	68.4	83	83.2	67.6	93.5	66.1	82.2	56.2	75

*Mapping to a *null* slot value is employed for all the unused slots in semantic frame.

falsely 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 applied to DOMOTICA 3 since all the slots are always inferred for each frame, hence no improvement is possible in this way (Despotovic et al., 2015).

Let us further assume a hierarchical learning approach where we first define MLN that learns mappings to slots, then subsequently for each slot define a separate MLN that learns mappings to its slot values. We can observe a slight decrease in *F*-scores for both datasets; however the major benefit is the fact that the large task can in this way be divided into smaller subtasks. This relaxes one of the main drawbacks of the MLNs, i.e., the fact that for large tasks the inference is potentially very slow (Kennington and Schlangen, 2014). There is also a benefit in terms of computational complexity in the learning phase: learning time is decreased by 27%. Results for different MLN setups are summarized in Tables 7 and 8 for datasets DOMOTICA 3 and PATCOR respectively.

6. Conclusions

Machine learning has received much attention in the spoken language understanding community in recent years. Our aim in this paper was to give a comparative analysis of a variety of state-of-the-art machine 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

Bellegarda, J.R., Monz, C., 2016. State of the art in statistical methods for lang. and speech processing. Comput. Speech Language 35, 163–184. http://dx.doi.org/10.1016/j.csl.2015.07.001.

Berger, A.L., Della Pietra, S.A., Della Pietra, V.J., 1996. A maximum entropy approach to natural language processing. Comput. Ling. 22 (1), 39–71.

Boulis, C., Ostendorf, M., 2005. Text classification by augmenting the bag-of-words representation with redundancy-compensated bigrams. International Workshop in Feature Selection in Data Mining. pp. 9–16.

Chang, C.C., Lin, C.J., 2011. LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. 2 (3). http://dx.doi.org/10.1145/1961189.1961199.

Christensen, H., Green, P., Hain, T., 2013. Learning speaker-specific pronunciations of disordered speech. 14th Annual Conference of the International Speech Communication Association (INTERSPEECH 2013), Lyon, France. pp. 1159–1163.

Coppola, B., Moschitti, A., Riccardi, G., 2009. Shallow semantic parsing for spoken language understanding. Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers. pp. 85–88.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. Learn. 20 (3), 273–297. http://dx.doi.org/10.1023/A:1022627411411.

Deoras, A., Tur, G., Sarikaya, R., D. Hakkani-Tür, D., 2013. Joint discriminative decoding of words and semantic tags for spoken language understanding. IEEE Trans. Audio Speech Lang. Process. 21 (8), 1612–1621. http://dx.doi.org/10.1109/TASL.2013.2256894.

Despotovic, V., Haeb Umbach, R., Walter, O., 2015. Semantic analysis of spoken input using Markov logic networks. 16th Annual Conference of the International Speech Communication Association (INTERSPEECH 2015), Dresden, Germany. pp. 1859–1863.

Doyle, P., Leeper, H., Kotler, A.-L., Thomas-Stonell, N., ONeill, C., Dylke, M.-C., Rolls, K., 1997. Dysarthric speech: a comparison of computerised speech recognition and listener intelligibility. J. Rehabil. Res. Dev. 34 (3), 309–316.

Duffy, J.R., 2012. Motor Speech Disorders: Substrates, Differential Diagnosis, and Management, Third edition. Mosby.

Gaspers, J., Cimiano, P., 2014. Learning a semantic parser from spoken utterances. IEEE International Conference on Acoustics, Speech and Signal Processing, (ICASSP 2014), Florence, Italy. pp. 3201–3205.

Gemmeke, J.F., Ons, B., Tessema, N., hamme, H.V., van de Loo, J., Pauw, G.D., Daelemans, W., Huyghe, J., Derboven, J., Vuegen, L., Broeck, B.V.D., Karsmakers, P., Vanrumste, B., 2013. Self-taught assistive vocal interfaces: an overview of the ALADIN project. 14th Annual Conference of the International Speech Communication Association (INTERSPEECH 2013), Lyon, France. pp. 2039–2043.

Kartsaklis, D., 2014. Compositional Distributional Semantics with Compact Closed Categories and Frobenius Algebras. Ph.D. thesis. University of Oxford, UK.

Kennington, C., Schlangen, D., 2014. Situated incremental natural language understanding using Markov logic networks. Comput. Speech Lang. 28 (1), 240–255. http://dx.doi.org/10.1016/j.csl.2013.06.004.

Khot, T., Balasubramanian, N., Gribkoff, E., Sabharwal, A., Clark, P., Etzioni, O., 2015. Exploring Markov logic networks for question answering. Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal. pp. 685–694.

Kilimci, Z.H., Ganiz, M.C., 2015. Evaluation of classification models for language processing. International Symposium on Innovations in Intelligent Systems and Applications (INISTA 2015), Madrid, Spain. pp. 1–8. http://dx.doi.org/10.1109/INISTA.2015.7276787.

Kok, S., Sumner, M., Richardson, M., Singla, P., Poon, H., Lowd, D., Wang, J., Domingos, P., 2009. The Alchemy System for Statistical Relational AI. Technical Report. Department of Computer Science and Engineering, University of Washington, Seattle, WA.

Lansford, K.L., Liss, J.M., Caviness, J.N., Utianski, R.L., 2011. A cognitive-perceptual approach to conceptualizing speech intelligibility deficits and remediation practice in hypokinetic dysarthria. Parkinsons Dis.,. Article ID 150962.

Lee, D.D., Seung, H.S., 1999. Learning the parts of objects by nonnegative matrix factorization. Nature 401, 788–791. http://dx.doi.org/10.1038/44565.

Lee, D.D., Seung, H.S., 2000. Algorithms for non-negative matrix factorization. Advances in Neural Information Processing 13 (NIPS 2000).

Lenci, A., 2008. Distributional semantics in linguistic and cognitive research. Riv. Ling. 20 (1), 1–31.

Mairesse, F., Gasic, M., Jurccek, F., Keizer, S., Thomson, B., Yu, K., Youngx, S., 2009. Spoken language understanding from unaligned data using discriminative classification models. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2009), Taipei. pp. 4749–4752. http://dx.doi.org/10.1109/ICASSP.2009.4960692.

Manning, C.D., Raghavan, P., Schtze, M., 2008. Introduction to Information Retrieval. Cambridge University Press.http://dx.doi.org/10.1017/CBO9780511809071.

McCallum, A., Nigam, K., 1998. A comparison of event models for Naive Bayes text classification. AAAI/ICML-98 Workshop on Learning for Text Categorization, Madison, WI, USA. pp. 41–48.

Mengistu, K.T., Rudzicz, F., 2011. Comparing humans and automatic speech recognition systems in recognizing dysarthric speech. Adv. Artif. Intell. 291–300.

Middag, C., 2012. Automatic analysis of pathological speech. Ph.D. thesis. Ghent University, Belgium.

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

Montague, R., 1970. Universal grammar. Theoria 36, 373–398.

Mori, R.D., Béchet, F., Hakkani-Tür, D., McTear, M., Riccardi, G., Tur, G., 2008. Spoken language understanding–interpreting the signs given by a speech signal. IEEE Signal Process. Mag. 25 (3), 50–58. http://dx.doi.org/10.1109/MSP.2008.918413.

Newman, M.E.J., Girvan, M., 2004. Finding and evaluating community structure in networks. Phys. Rev. E 69 (2). http://dx.doi.org/10.1103/PhysRevE.69.026113.

Nigam, K., Laferty, J., McCallum, A., 1999. Using maximum entropy for text classification. IJCAI-99 Workshop on Machine Learning for Information Filtering, Stockholm, Sweden. pp. 61–67.

Ons, B., Gemmeke, J., Van hamme, H., 2014. The self-taught vocal interface. EURASIP J. Audio Speech Music Process. 43. http://dx.doi.org/10.1186/s13636-014-0043-4.

Ons, B., Gemmeke, J.F., Van hamme, H., 2013. NMF-based keyword learning from scarce data. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU 2013), Olomouc, Czech Republic. pp. 392–397. http://dx.doi.org/10.1109/ASRU.2013.6707762.

Ons, B., Tessema, N., van de Loo, J., Gemmeke, J., Pauw, G.D., Daelemans, W., hamme, H.V., 2013. A self learning vocal interface for speech-impaired users. 4th Workshop on Speech and Language Processing for Assistive Technologies (SLPAT), Lyon, France. pp. 78–81.

Poon, H., Domingos, P., 2006. Sound and efficient inference with probabilistic and deterministic dependencies. The 21st National Conference on Artificial Intelligence (AAAI '06), Boston, Massachusetts, USA. pp. 458–463.

Porter, F.C., Narsky, I., 2013. Statistical Analysis Techniques in Particle Physics. Wiley.

Richardson, M., Domingos, P., 2006. Markov logic networks. Mach. Learn. 62 (1–2), 107–136.

Sanders, E., Ruiter, M.B., Beijer, L., Strik, H., 2002. Automatic recognition of dutch dysarthric speech: a pilot study. 7th International Conference on Spoken Language Processing, ICSLP2002 - INTERSPEECH 2002, Denver, Colorado, USA.

Singh, B., Rani, V., Mahajan, N., 2012. Preprocessing in ASR for computer machine interaction with humans: a review. Int. J. Adv. Res. Comput. Sci. Softw. Eng. 2 (3), 396–399.

Sutton, C., McCallum, A., 2011. An introduction to conditional random fields. Found. Trends Mach. Learn. 4 (4), 267–273. http://dx.doi.org/10.1561/2200000013.

Taskar, B., Abbeel, P., Wong, M., Koller, D., 2007. Relational Markov Networks. In: Getoor, L., Taskar, B. (Eds.), Introduction to Statistical Relational Learning. MIT Press, chapter 6, pp. 175–199.

Tessema, N., Ons, B., van de Loo, J., Gemmeke, J., De Pauw, G., Daelemans, W., Van hamme, H., 2013. Metadata for Corpora PATCOR and Domotica-2. Technical Report. KU Leuven.

Tur, G., De Mori, R., 2011. Spoken Language Understanding: Systems for Extracting Semantic Information from Speech. John Wiley & Sons Ltd.

Wallach, H.M., 2004. Conditional Random Fields: An Introduction, Technical Report MS-CIS-04-21. Technical Report. Department of Computer and Information Science, University of Pennsylvania.

Walter, O., Despotovic, V., Haeb-Umbach, R., Gemmeke, J., Ons, B., Van hamme, H., 2014. An evaluation of unsupervised acoustic model training for a dysarthric speech interface. 15th Annual Conference of the International Speech Communication Association (INTERSPEECH 2014), Singapore. pp. 1013–1017.

Walter, O., Korthals, T., Haeb-Umbach, R., Raj, B., 2013. Hierarchical System for Word Discovery Exploiting DTW-Based Initialization. Automatic Speech Recognition and Understanding Workshop (ASRU 2013). pp. 386–391.

Wang, Y., Deng, L., Acero, A., 2011. Semantic Frame-based Spoken Language Understanding. In: Tur, G., Mori, R.D. (Eds.), Spoken Language Understanding: Systems for Extracting Semantic Information from Speech. Wiley, pp. 41–91. http://dx.doi.org/10.1002/9781119992691.ch3.

Wang, Y.Y., 2010. Strategies for statistical spoken language understanding with small amount of data - an empirical study. 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010), Makuhari, Chiba, Japan. pp. 2498–2501.

Wang, Y.Y., Acero, A., 2006. Discriminative models for spoken language understanding. International Conference on Spoken Language Processing, Pittsburgh, PA, USA. pp. 1766–1769.

Weilin, W., Ruzhan, L., Liu, Z., 2003. Comparative experiments on task classification for spoken language understanding using Naive Bayes classifier. International Conference on Natural Language Processing and Knowledge Engineering, Beijing, China. pp. 492–497. http://dx.doi.org/10.1109/NLPKE.2003.1275955.

Weinman, J., Lidaka, A., Aggarwal, S., 2011. Large-scale Machine Learning. In: mei W. Hwu, W. (Ed.), GPU Computing Gems Emerald Edition. Morgan Kaufmann Publishers, pp. 277–291. http://dx.doi.org/10.1016/B978-0-12-384988-5.00019-X.

Wutiwiwatchai, C., Furui, S., 2003. Confidence scoring for ann based spoken language understanding. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU 2003), St. Thomas, U.S. Virgin Islands. pp. 566–571. http://dx.doi.org/10.1109/ASRU.2003.1318502.