Wang and Van hamme
EURASIP Journal on Audio, Speech, and Music Processing (2023) 2023:15
https://doi.org/10.1186/s13636-023-00280-z

EURASIP Journal on Audio, Speech, and Music Processing

EMPIRICAL RESEARCH Open Access

Benefits of pre-trained mono- and cross-lingual speech representations for spoken language understanding of Dutch dysarthric speech

Pu Wang* and Hugo Van hamme

Abstract

With the rise of deep learning, spoken language understanding (SLU) for command-and-control applications such as a voice-controlled virtual assistant can offer reliable hands-free operation to physically disabled individuals. However, due to data scarcity, it is still a challenge to process dysarthric speech. Pre-training (part of) the SLU model with *supervised* automatic speech recognition (ASR) targets or with *self-supervised* learning (SSL) may help to overcome a lack of data, but no research has shown which pre-training strategy performs better for SLU on dysarthric speech and to which extent the SLU task benefits from knowledge transfer from pre-training with dysarthric acoustic tasks. This work aims to compare different mono- or cross-lingual pre-training (*supervised* and *unsupervised*) methodologies and quantitatively investigates the benefits of pre-training for SLU tasks on Dutch dysarthric speech. The designed SLU systems consist of a pre-trained speech representations encoder and a SLU decoder to map encoded features to intents. Four types of pre-trained encoders, a mono-lingual time-delay neural network (TDNN) acoustic model, a mono-lingual transformer acoustic model, a cross-lingual transformer acoustic model (Whisper), and a cross-lingual SSL Wav2Vec2.0 model (XLSR-53), are evaluated complemented with three types of SLU decoders: non-negative matrix factorization (NMF), capsule networks, and long short-term memory (LSTM) networks. The acoustic analysis of the four pre-trained encoders are tested on Dutch dysarthric home-automation data with word error rate (WER) results to investigate the correlations of the dysarthric acoustic task (ASR) and the semantic task (SLU). By introducing the intelligibility score (IS) as a metric of the impairment severity, this paper further quantitatively analyzes dysarthria-severity-dependent models for SLU tasks.

Keywords Spoken language understanding, Low resources dysarthric speech, Pre-training, Self-supervised learning, Transformers, Time-delay neural network, Whisper, XLSR-53, Wav2Vec2, Impairment intelligibility

1 Introduction

Spoken language is a natural way for people to interact with others. Nowadays, one is also able to communicate with devices by voice. For example, one can speak to their virtual assistant to set alarms or play music. Voice-controlled devices offer hands-free operation, replacing keyboards, mice, and touch screens, which is more friendly and accessible to physically challenged individuals or elderly lacking fine motor skills. In conjunction with domestic devices, vocal assistive technology can help their users to live more independently. Operations like controlling lights or setting the heating temperature can be completed by uttering simple commands like “turn on/off the reading light” or “set the heating to 23

*Correspondence:
Pu Wang
pu.wang@esat.kuleuven.be
Department of Electrical Engineering-ESAT, KU Leuven, Leuven, Belgium

degrees" to a vocal user interface without even pressing any button.

The cause of loss of (fine) motor skills often also leads to speech impairments [1]. Assistive technology is hence beneficial for the quality of life for physically challenged users, however, [2] observes that current voice assistants only reach on average an accuracy of 50–60% on impaired speech while the minimal satisfactory rate is regarded as 90–95%. The primary goal of this manuscript is to design a dysarthric speech vocal assistance system.

A key component of vocal assistive technology is a spoken language understanding (SLU) system. A SLU system is built to extract semantics from the spoken commands and map them to the target task. Taking the aforementioned command "turn on the light" as an example, the SLU task considered here can be described as recognizing an intent (i.e., to manipulate the lights rather than, e.g., asking for the time) and assigning a value to two relevant slots: slots *action* and *object* should be assigned the values *state on* and *light*. Therefore, we formulate the goal of the SLU system as the multi-class mapping of the spoken command to a set of labels representing the intents and discrete slot values.

Traditional approaches use a pipeline structure consisting of two modules [3]: an automatic speech recognition (ASR) module that converts the spoken command to a textual transcription, and subsequently a natural language understanding (NLU) component that extracts meaning from the text. Designing such a system can be challenging for dysarthric speech. First of all, ASR performs poorly on impaired speech [4, 5]. Though state-of-the-art ASR models with deep learning approaches have advanced greatly, they are challenged by the data scarcity issue when it comes to dysarthric speech [6]. Recording speech from individuals with dysarthria is hindered by greater recruitment efforts as well as speaker exhaustion. Moreover, vocal characteristics vary greatly between impaired speakers and impairment types [6]. Therefore, the lack of data and the large speech variation lead to a drastic drop in ASR performance. For instance, [7] observes that Google's speech recognition system does not work well for speech produced by deaf users.

The poor ASR accuracy creates a significant mismatch when the ASR and NLU components are designed and trained separately. When the NLU component is trained with clean transcriptions, it is not robust to handle errors generated by the ASR component. Also, the ASR component is trained to minimize errors between hypothesis and reference (or a proxy thereof), but not all words are equally important for the cascaded NLU component [8]. End-to-end (E2E) SLU systems avoid the two-step procedure by mapping speech directly onto semantics without using an intermediate text representation [9]. Recently, an increasing interest in this approach to SLU is witnessed (e.g., [10]). For dysarthric speech input, E2E SLU has the additional advantage that there is no explicit ASR component involved. Indeed, for disordered speech, the ASR component will generate many errors disrupting two-step SLU systems. Multiple strategies for adapting an ASR model for dysarthric speech input are discussed in [11], but there is no "one-fits-all" solution. In prior related work, we have hence opted to avoid explicit ASR in SLU.

1.1 Related work

We proposed to build a vocal assistant that is fully trained from scratch by end-users' spoken commands and accompanying demonstrations encoded as a semantic frame as explained above [12–15]. The SLU system will learn the direct mapping from the utterance to the intent labels. To capture idiosyncrasies in voice disorder as well as in linguistic habits, the SLU system is typically trained in a speaker dependent setting. The training samples therefore avoid the high inter-speaker variations of disordered speech and are sufficiently consistent and discriminative. Since the demonstrations come from end-users, the system will be matched to the end-users' way of formulating intents. Also, the mapping from utterances to semantics is direct without intermediate transcription. The SLU system will hence not degrade by feeding the NLU system with corrupted ASR output for dysarthric speech. The challenge is that the approach needs training samples and teaching time from its end-users. To minimize the users' efforts, the designed algorithms are expected to quickly converge after only a few training samples. In previous research, we have investigated three frameworks for this goal.

We start from a non-negative matrix factorization (NMF) model to find recurring patterns in the inputs and link them to semantic labels. Previous work [9, 12] has shown that NMF is effective for learning the meaning of a limited vocabulary for simple commands. However, NMF is a bag-of-words model and therefore insensitive to word order (grammar).

Since the utterance representation does not adhere exactly to the linear NMF model, we replaced this utterance decoder with a capsule network [16]. The capsule network models the hierarchical relation in speech (phones form words, which form a command) [13, 14] and can capture sequential information through an RNN encoder [17]. Although, compared to the former NMF-based frameworks, capsule networks are more powerful when more training data is available [13], the NMF model outperforms the capsule network on small training sets, which may be caused by the fact that the RNN encoder suffers from the inconsistent acoustic patterns more than NMF.

To alleviate the limitations caused by the RNN, we further propose to utilize a variant of the transformer structure to replace the RNN, which mainly adopts the self-attention mechanism to model relations between elements in the sequence and therefore is much more efficient [15, 18]. Although this method produces convincing results when dealing with dysarthric speech compared to previous work, as explained in [15], it still fails to obtain the desired performance levels when it comes to only a few tens of training samples.

1.2 Pre-training methodologies

For SLU on typical speech, [10, 19–24] show that incorporating representations from pre-trained acoustic models, instead of using the typical filterbank or MFCC features, leads to better performance. Bhosale et al. [25] verified this idea in a speaker-independent setting by pre-training two acoustic models on corpora of typical 1000 h of Librispeech data with ASR targets and extracting several layers from the acoustic model as the pre-trained layers in the SLU model. The SLU model achieved up to 25% performance gain compared to the baseline model with filterbank features as inputs on a Dutch dysarthric dataset, which proves layers extracted from pre-trained ASR models on typical speech can significantly boost the performance of a dysarthric speech SLU model. Since pre-training is a data-driven technique to alleviate data scarcity in scenarios where training and testing data have the same distribution, in this manuscript, we will extend this study and investigate whether pre-training on dysarthric corpora will help to improve the dysarthric speech SLU performance compared to pre-training on corpora with typical speech. The aforementioned investigations involve a supervised training step on dysarthric speech, which requires both the speech audio and its corresponding transcriptions to be available. While accurate transcriptions are not necessary for SLU tasks, preparing larger-scale data transcriptions is tedious, especially for dysarthric speech. A recent method to alleviate the need for labeled data is self-supervised representation learning, where masked or future data are predicted from available data. SSL speech models, such as Wav2Vec2.0 [26], and Hubert [27], have already been applied to a variety of speech tasks including ASR[26], emotion recognition [28], speaker identification [29], and phoneme classification [30]. Hernandez et al. [31] report that both cross-lingual SSL speech representations, XLSR-53, and the mono-lingual Wav2Vec2.0 outperform the filterbank features on English, Spanish, and Italian dysarthric speech ASR. However, it remains an open question whether better ASR is the key to improving downstream SLU since the former task relies on high-level acoustic representations while the latter task is highly related to semantics. Peng et al. [32] observe that for SLU on typical speech, SSL speech models yield better performance gains than supervised pre-trained models. Therefore, in this manuscript, we further analyze if the speech representations learned with SSL offer a better alternative as input for downstream dysarthric SLU tasks than the aforementioned supervised acoustic models.

1.3 Contributions

Work on improving ASR and speech representations for dysarthric speech does not automatically carry over to E2E SLU systems, because this setting assumes task-specific data will be available. What is the best architecture and training strategy in this context? Do better acoustic representations for ASR contribute to better SLU performances? How much does pre-training on target speech help? In this manuscript, we report on four aspects.

First, pre-training on dysarthric speech can be challenging. The less consistent and wider data distribution hassle the acoustic model and speech representations to describe dysarthric speech in a generalizable way and undermines the possibility for knowledge transfer to other tasks. Previous studies demonstrated bottleneck features (BNF) yield stable representations of dysarthric speech [33, 34], while [35] shows improvements in dysarthric ASR by exploring a time-delay neural network (TDNN). Inspired by these, we explore pre-training of a TDNN acoustic model on a publicly available dysarthric speech corpus with ASR targets and then extract layer activations of the well-trained TDNN model as BNFs for dysarthric end-user utterances. The TDNN-based acoustic model is evaluated with capsule networks in [36] to predict SLU intent labels. To verify that we are learning something from the TDNN-based acoustic model other than just the idiosyncrasies of a particular decoding model, we will extensively evaluate it with three types of SLU decoders for semantic inference: the NMF model, the multilayer capsule network model, and the LSTM model. Cross-validation experiments are performed to measure the effect of pre-training on SLU performance on a corpus of dysarthric speech. Variance estimations will be applied to test the statistical significance of the differences observed. Besides the TDNN, the transformer network is a popular more recent neural network structure which has been widely used in language representation tasks but which fails under limited training samples as tested in our previous work [15]. Since larger-scale transformer-based models such as BERT [37], ALBERT [38], HuBERT [27], the ERNIE framework [39], XLnet [40], the Wav2Vec2.0 framework [26], and XLS-R [41] have shown to be effective in various language understanding domains when pre-training is involved, we will again explore the transformer structure for dysarthric speech SLU application. As discussed in the Section 1.2, features extracted from the transformer-based SSL Wav2Vec2.0 model have shown promising results for dysarthric speech ASR. Therefore, we build both a *supervised* transformer-based acoustic model and a Wav2Vec2.0 *self-supervised* representation learning model. The pre-trained features extracted from these two models will also be tested with three types of SLU decoders and the results will be compared to the TDNN-based acoustic model to explore which is the best architecture and pre-training strategy for this context.

Second, the relations between *acoustic* representation learning and the *semantic* SLU task are unclear. Besides the semantic (SLU) accuracy, acoustic analysis with word error rate (WER) results will also be conducted in this work. To better analyze how knowledge transfers from the task-agnostic corpus to the task-specific data, all the pre-trained encoders will be tested under the zero-shot setting. More specifically, all pre-trained encoders will be first fine-tuned on a task-agnostic dysarthric corpus. The well-trained encoders will be cascaded by a simple connectionist temporal classification (CTC) decoder with beam search for ASR inference on the unseen SLU task-specific data.

Third, the limited size of existing dysarthric speech databases does not allow to train a large-scale acoustic model from scratch. To augment the size of the pre-training dataset, Vachhani et al. [42] uses temporal and speech modifications to typical speech to generate synthetic speech that matches the characteristics of dysarthric speech. In this manuscript, instead of directly adding typical or pseudo-disordered speech to a dysarthric corpus which may lead to data imbalance issues, we follow a two-stage pre-training strategy. Our acoustic model will be initialized by training on larger-scale typical Dutch speech data and will then be fine-tuned on a mixture of Dutch dysarthric speech and the same size of typical speech. This idea is supported by [43]. Besides training an encoder with the target language, large-scale cross-lingual models, pre-trained on multiple languages or multiple speech tasks show more robust and better-generalizing representations, for example, Takashima et al. [5] separately pre-trains typical and dysarthria-specific acoustic models and then joins them by introducing multilingual typical speech to the dysarthric speech dataset. Hernandez et al. [31] shows that features extracted from the multilingual model XLSR-53, which is trained on 56,000 h of audio from 53 different languages, led to lower WERs than models trained on a single language. We hereby introduce the multilingual multitask acoustic model Whisper [44] and the SSL XLSR-53 [45] model as the pre-trained encoders to explore if cross-lingual speech representations can advance the Dutch dysarthric speech SLU. Moreover, the investigation of the cross-lingual implementations in which language properties are not explicitly exploited suggests our results hold for dysarthric SLU in other languages than Dutch, though we will not formally evaluate this.

Fourth and finally, since dysarthric speech varies greatly between speakers with different impairment severity, there is a concern that the knowledge learned by pre-training on specific speakers may not generalize to other speakers. [46–48] demonstrate that speaker adaptation is useful for dysarthric speech ASR. They propose to classify the dysarthric speakers into predefined severity levels at first and then select the data sources for training based on different severity levels. However, considering the efforts introduced by evaluating impairment severity in advance and pre-training acoustic models separately, the benefits of constructing such dysarthria-severity-dependent acoustic models for SLU tasks are unclear since the optimization goal of SLU systems is semantics instead of word error rate and its training process will learn to correct ASR errors. In this manuscript, we will use the speaker's intelligibility score (IS) as an approximate metric of the impairment severity for both the pre-training corpus and the evaluation utterances to discuss the extent to which SLU performance is influenced by impairment severity. Therefore, the acoustic model will be trained on disordered speech collected from different IS ranges to yield several independent pre-trained models.

The rest of the paper is structured as follows. In Section 2, we will describe the SLU methods included in the comparison in detail. The structure and pre-training procedure of (mono-lingual supervised) TDNN-based acoustic model, (mono-lingual supervised) transformer-based acoustic model, (cross-lingual supervised) Whisper-based acoustic model, and (cross-lingual self-supervised) XLSR-53-based speech model will be given. The related NMF-based, capsule network-based, and LSTM-based SLU decoder will also be covered. In Section 3, we will discuss the experimental methodology as well as the data sets that are used in the SLU tasks. In Section 4, we will analyze the four pre-trained encoders from both the acoustic (WER on ASR task) and semantic (SLU accuracy) perspective while involving task-agnostic dysarthric speech. Section 5 will cover the effect of pre-training acoustic models

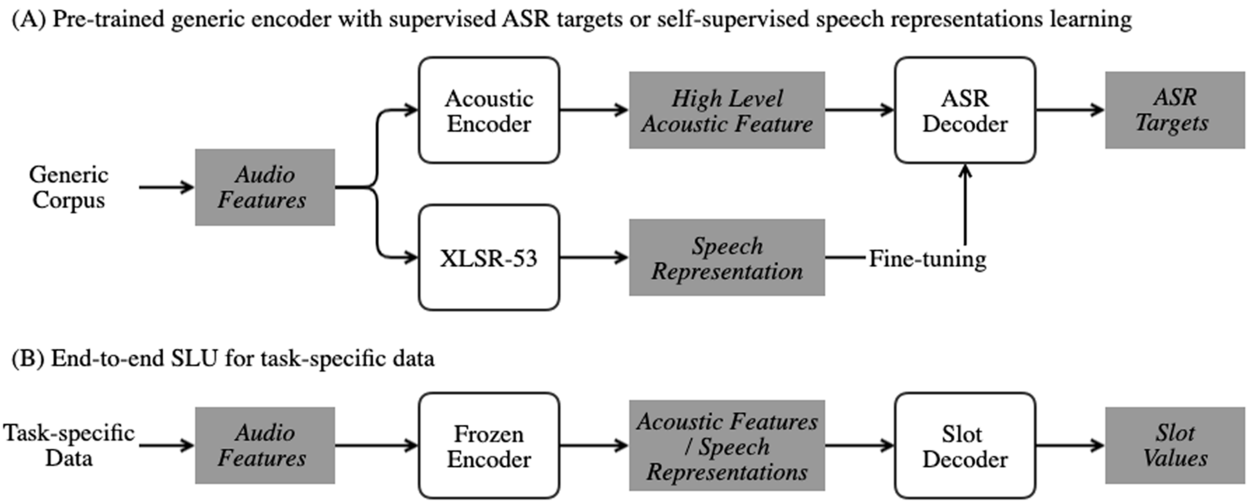

(A) Pre-trained generic encoder with supervised ASR targets or self-supervised speech representations learning
Generic Corpus
Audio Features
Acoustic Encoder
High Level Acoustic Feature
ASR Decoder
ASR Targets
XLSR-53
Speech Representation
Fine-tuning
(B) End-to-end SLU for task-specific data
Task-specific Data
Audio Features
Frozen Encoder
Acoustic Features / Speech Representations
Slot Decoder
Slot Values

Fig. 1 Overall structure of the SLU system with pre-trained acoustic speech representations

with dysarthric speech collected from selected impairment severity. Finally, Section 6 will conclude the work.

Selected fine-tuned acoustic models[1] and decoder implementations[2] are publicly available.

2 Models and data

We consider encoder-decoder models as shown in Fig. 1. The encoder maps speech onto high-level acoustic speech representations that facilitate a low complexity decoder. The outputs of the decoder are task-specific slot value activations; the decoder is therefore trained on task-specific data. To allow for pre-training, the encoder is generic, i.e., trained on task-agnostic typical and/or disordered speech. As shown in Fig. 1(A), the acoustic encoders are trained *supervisedly* for the ASR task, while the speech representation model is *self-supervised* learning by predicting masked frames from the unmasked ones using a contrastive loss. To better compare these two pre-training strategies, the SSL XLSR-53 model is further fine-tuned with a CTC decoder on an ASR task. The detailed procedure will be discussed in the Section 2.2.2. The well-trained encoder is then frozen and the decoder is trained with a loss function that is a proxy for accuracy, such as maximal cross-entropy between inferred slot value activations and their ground truth or data likelihood (Fig. 1(B)). We opt not to adapt the encoder on the task-specific data for this might degrade its performance on future SLU tasks, e.g., when more functionality is trained by the end user.

In this section, we will first discuss four implementations of the encoder. The first implementation is a mono-lingual acoustic model built with a transformer network structure and will be referred to as the transformer-based acoustic model. The second one is also a mono-lingual model built with TDNN/TDNN-F and will be referred to as the TDNN-based acoustic model. The third one is a cross-lingual acoustic model built with the Whisper framework [44] and will be referred to as the Whisper-based acoustic model. The final one is a cross-lingual SSL model built with the XLSR-53 [45] framework and will be referred to as the XLSR-53-based speech model. Below, we will detail the pre-training process as well as the task-agnostic training corpora. Finally, we will introduce three types of intent decoders which are built using NMF, capsule networks and LSTMs respectively, as well as their implementations for our dysarthric speech SLU tasks.

2.1 Mono-lingual encoders

We first show the two mono-lingual acoustic models in this section.

2.1.1 Transformer-based acoustic model

The transformer is a layer-stacked neural network for modeling sequence data [18]. Each layer transforms the sequence into a new sequence of the same length using two sub-layers, the multi-head attention layer and the fully connected feed-forward layer as shown in Fig. 2. At each position in the sequence, each head makes a weighted average of the feature vectors in the previous layer. The weights are determined in the attention layer through the inner product of Key and Query vectors, which are linear transformations of the sequence. The attention in a head hence looks for matching data

[1] https://huggingface.co/kul-speech-lab

[2] https://github.com/wangpuup/assist_dy

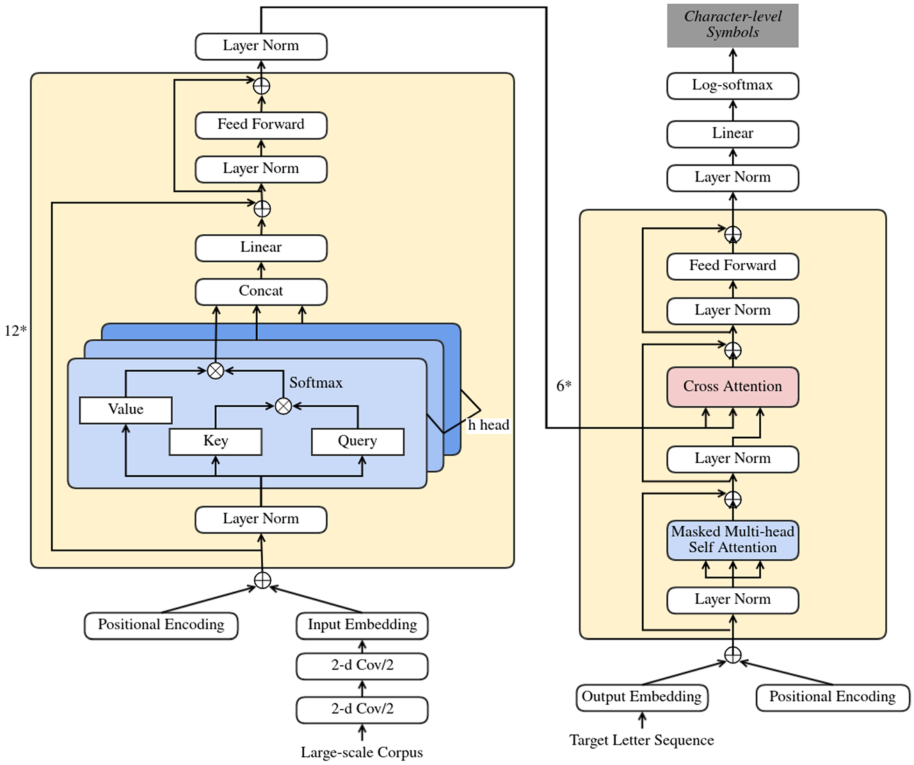


Fig. 2 The structure of the transformer-based acoustic model

properties in the sequence. Each head can specialize in different properties.

Mathematically,

$$Attn^i = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

$$MultiHead = W_{out}Concat(Attn^1, Attn^2, ...Attn^N) \tag{2}$$

where $Attn^i$ is the expression of the ith attention head. Q, K, and V are Query, Key, and Value respectively for the ith head. W_{out} is the weight matrix combining the outputs of the different heads. This result is then transformed through the fully connected feed-forward layer and forwarded to the next transformer layer.

The encoder is trained for ASR targets jointly with a transformer-based auto-regressive decoder. The decoder has the same structure as the encoder, except a multi-headed cross-attention block is inserted in every layer, where Key and Value come from the high-level features of the encoder and Query comes from the self-attended decoded sequence in the decoder. To prevent self-attention from attending to future positions in the decoder, a mask is applied as well [49].

The transformer acoustic model is first trained with ASR targets by taking feature sequences of audio from a (large-scale) task-agnostic corpus as the inputs and their related character-level symbol sequences as the target outputs. The input sequences will be first enriched by a position embedding to include order information. Since the memory consumption of the transformer structure grows quadratically with the sequence length, a 2-layer 2-dimensional convolutional layer with kernel size (3,3) and 256 channels implements a 4-fold sub-sampling along the time dimension of the input sequences. In our implementation, the transformer encoder is composed of a stack of 12 identical layers and the decoder is composed of 6 layers. Each identical layer has a 4-head 256-dimensional attention layer and a 2048-dimensional feed-forward layer.

The transformer acoustic model is built and trained with the ESPnet toolkit in an end-to-end manner by employing the multi-objective learning framework [50].

2.1.2 *TDNN-based acoustic model*

A TDNN is a multilayer feed-forward model with temporal splicing (frame splicing) in the internal layers of the feed-forward structure. A deep TDNN architecture can

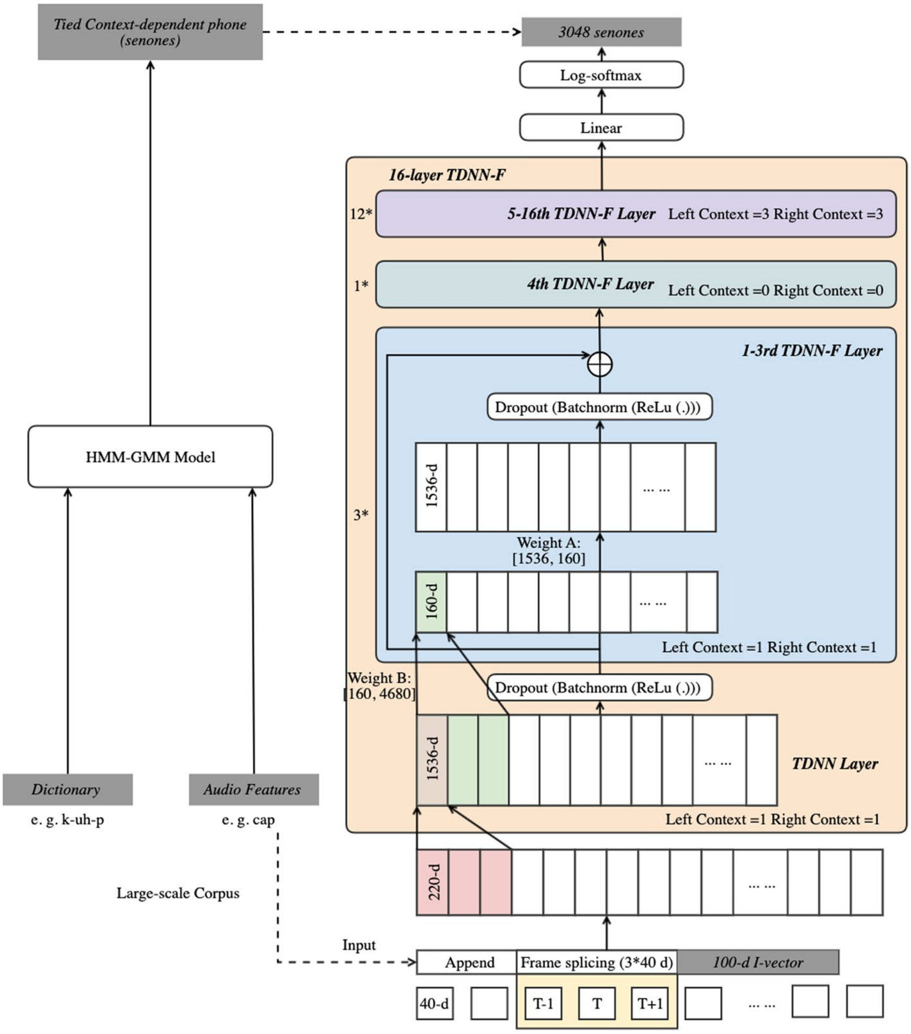


Fig. 3 The structure of the TDNN-based acoustic model

capture long-term dependencies between acoustic events and has been shown to be effective in ASR [51]. As shown in Fig. 3, the initial TDNN layer is learnt on narrow contexts. For example, using a left and right context of one frame, the input is the splicing of three context frames (the left frame - the current frame - the right frame). The next layer with another left and right context of one frame will have a receptive field of two frames to the left and two frames to the right. Therefore, the deeper layers have the ability to learn wider temporal relationships. TDNN-F is a factorized TDNN proposed by [52]. It uses singular value decomposition to factorize each learned weight matrix of a TDNN layer as a product of two much smaller factors and forces one of the two factors to be semi-orthogonal to reduce the number of parameters. The most direct application of this idea is to introduce one linear bottleneck layer between the two TDNN layers as shown in the blue block of Fig. 3. Suppose a typical TDNN topology with a hidden layer dimension of 1536. The weight matrix would have a shape of [1536, 4608], where 4608 corresponds to the 3 frame offsets of the previous layer spliced together. By introducing the linear bottleneck layer with a smaller interior dimension of, for example, 160, the weight matrix is factorized into weights A of size [1536, 160] and weights B of size [160, 4608], with B constrained to be semi-orthogonal. This operation is shown to be helpful in both performance and efficiency [52].

We built the TDNN acoustic model with the Kaldi toolkit [53] as shown in Fig. 3. TDNN training involves a stage in which similar context-dependent HMM emission models are clustered into so-called senones. The alignments between the input feature frames and senones are generated by the HMM-GMM model. We take 40-dimensional MFCC features as the inputs with cepstral mean and variance normalization (CMVN) applied.

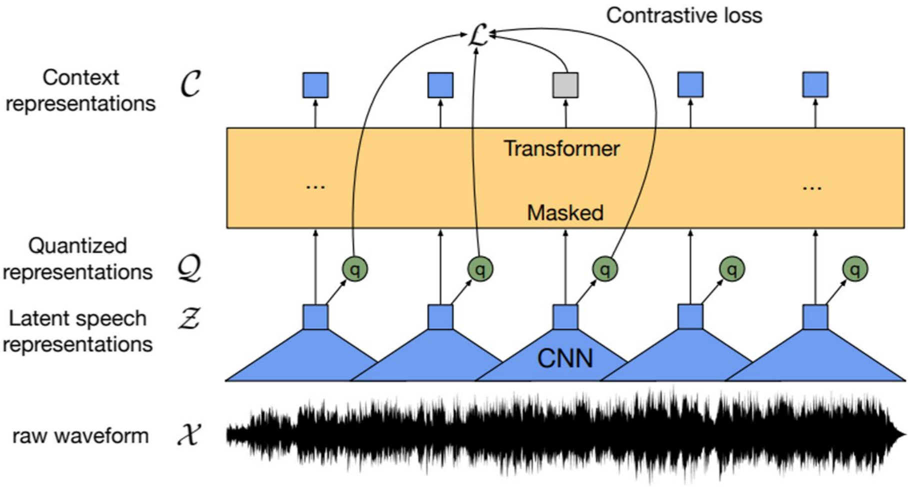


Fig. 4 Structure of the Wav2Vec2.0-based speech model [26]

Since dysarthric speech varies greatly between speakers, it is important for the acoustic model to capture and learn relevant information about the speaker; otherwise, the knowledge learned from the pre-trained dysarthric speakers would fail to transfer to the target speakers in the subsequent SLU tasks. We therefore concatenate 100-dimensional I-vectors to the MFCC features. In our implementation, the TDNN model is composed of a single layer TDNN followed by a 16-layer TDNN-F. The single layer TDNN has a 1536-dimensional hidden layer using a left and right context of 1 frame. The linear bottleneck layer of each layer in the 16-layer TDNN-F has a factorization dimension of 160. The hidden layers of the TDNN-F have the same dimension as the TDNN. The first 3 layers of the TDNN-F splices 1 frame of the left and of the right context of the previous layer as the input while the 4th TDNN-F layer only focuses on the current frame. The last 12 layers splice 3 frames of the left and the right context. Therefore the TDNN model uses a left and right context of 40 frames (4*1 frame + 12*3 frames) in total.

After completing training with ASR targets, the parameters of the well-trained acoustic model are frozen. The 160-dimensional bottleneck layer extracted from the 16th layer of the TDNN-F will serve as the feature encoder in the end-to-end dysarthric speech SLU systems.

2.2 Cross-lingual encoders

We will discuss the cross-lingual acoustic model and SSL speech representation learning model in this section.

2.2.1 Whsiper-based acoustic model

Whisper is a pre-trained model for ASR released by OpenAI [44]. It is trained using supervised learning on 680, 000 hours of multilingual audio and transcription data in 96 languages. This results in extensive acoustic knowledge that can be applied to over 96 languages. Even on languages considered low-resourced, results competitive to state-of-art ASR are achieved. As reported by Radford et al. [44], through fine-tuning, the pre-trained acoustic model can be adapted for specific datasets and languages to yield a performant ASR. In this manuscript, we therefore fine-tune the pre-trained Whisper encoder with task-agnostic Dutch typical and dysarthric speech.

The Whisper architecture is an implementation of an encoder-decoder transformer as shown in Section 2.1.1 (Fig. 2). It maps a sequence of audio features to a sequence of text tokens. An important difference is Whisper is trained using multi-task learning with tasks that include transcription, translation and timestamp prediction. It, therefore, generates informative latent features other than solely acoustic latent features with a transformer-based acoustic model. This might yield a benefit in SLU tasks.

The Whisper checkpoints come with five configurations of varying model sizes. We will fine-tune the multilingual version of the *small* checkpoint[3] (12-layer transformer encoder with twelve 768-dimensional heads). After fine-tuning the checkpoint with the Dutch corpus, the pre-trained acoustic features for the downstream SLU task are extracted from the last projection layer of the encoder.

[3] https://huggingface.co/openai/whisper-small

2.2.2 XLSR-53 speech model

XLSR-53 is a cross-lingual variant of Wav2Vec2.0 which is trained on with 56,000 h of unlabeled speech in 53 languages.

Figure 4 shows a typical Wav2Vec2.0 model [26]. It first takes raw waveforms as input to a feature encoder (temporal CNN blocks) to get the latent features. The latent features are then quantized. The quantized features are masked randomly and fed to a transformer-based context network which is trained to predict the masked features. The training process depends on a contrastive loss in which the model needs to identify the true (masked) quantized features.

Though Wav2Vec2.0 yields high-quality acoustic representations of speech, one cannot expect it to learn a speech to semantics mapping. Matsushima [54] therefore proposed to divide the Wav2Vec2.0 training into two phases, pre-training and fine-tuning. In the pre-training phase, the model is continually trained on task-agnostic typical speech. In the fine-tuning phase, a linear projection is added on top of the contextualization network, which is updated from an ASR CTC loss with the labeled task-agnostic dysarthric speech.

Matsushima investigated the LARGE XLSR-53 model[4] (which consists of 24 transformer blocks with sixteen 1024-dimensional heads) on zero-shot Dutch dysarthric speech ASR. The WER results are compared to supervised ASR training. In this manuscript, we will extend the size of the tested corpus and further extract the representations from the XLSR (transformer-based) context network fed to the SLU decoders.

2.3 Pre-training acoustic speech models

Existing dysarthric speech corpora, for example, the Universal Access (UA) speech corpus [55], TORGO [56], Nemours [57], etc., are substantially smaller than speech corpora of typical speech used for (pre-)training. To overcome this issue of unavailability of suitable speech data, we pre-train the acoustic/speech model in two stages. In the first stage, we either initialize or continually train the acoustic/speech model on a corpus of typical speech. In the next stage, we fine-tune a part of the pre-trained model by joint training on the mixture of a dysarthric speech corpus and the typical speech corpus from the first stage.

2.3.1 Pre-training corpora

The data used for pre-training originates from two corpora.

Table 1 Statistics of the Copas corpus

Severity(IS)	# of speakers	# of hours
Mild (> 85)	99	1.95
Moderate (70–85)	63	1.41
High (60–70)	8	0.2
Severe (< 60)	12	0.4
Total	182	3.96

Corpus Gesproken Nederlands (CGN) [58] is a corpus of typical Dutch speech as spoken in Flanders and Netherlands. It contains 14 components including read speech, broadcast comments, interviews, conversations, and telephone dialogues. We use data from the Flemish part of 11 components excluding the component *a* (spontaneous conversations) and components *c* and *d* (narrow-band recordings). The training data is composed of 138297 utterances containing 76115-word forms, which is about 133 h in total.

Copas [59] is a Dutch corpus of pathological speech recorded in Flanders. A total of 183 speakers have performed the Dutch intelligibility assessment (DIA) [60], resulting in an intelligibility score (IS) which will be used as a metric of speech disorder severity. This score further can be estimated automatically by [61].

Since the transcription is required for ASR targets in pre-training, we only use the audio collected from 5 components: DIA (speakers read 50 consonant-vowel-consonant words for phoneme-level intelligibility assessment), S1 (speakers imitate a spoken example of one sentence with the appropriate intonation and stress), S2 (speakers repeat one sentence after the spoken example), T (speakers read 11 different text passages with reading difficulty level AVI 7 or 8), and TM (speakers read a standardized (phonetically balanced text) which are provided with both the orthographic transcriptions (the target texts that are read) and the transliteration of the speech (the actual text perceived by the annotator). We use the transliteration of the speech for training. It contains 10,792 utterances with 1160 word forms, of which 575 words occur in CGN. We summarize the speaker information and IS in Table 1. The speech recordings are divided into 4 severity levels based on the IS. The highest IS is 100, i.e., typical speech, and the lowest score is 28 which is considered as severely impaired.

2.3.2 Pre-training strategy

The pre-training procedures for the four encoders are organized in two stages. The pre-training/fine-tuning for the XLSR-53-based speech model has been explained in Section 2.2.2, so we will only discuss three acoustic models here.

[4] https://github.com/Tatsu1020/self-supervised-dutch-dysarthria-asr

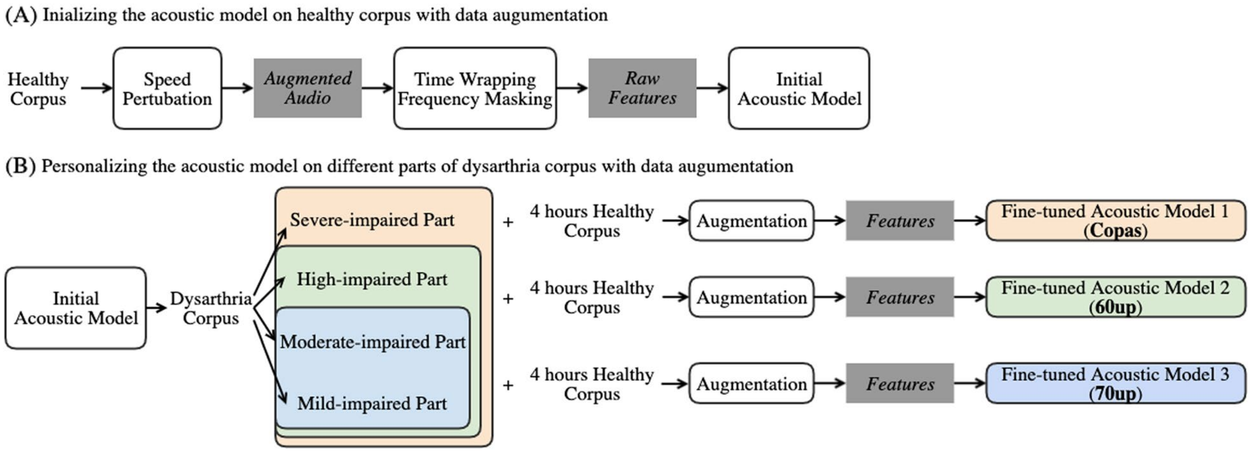


Fig. 5 The two-stage pre-training acoustic models

As shown in Fig. 5(A), we first build an initial acoustic model trained on the CGN corpus containing typical speech only. For the transformer-based model, the initial acoustic model is trained to map to 67 character-level symbols. For the TDNN-based model, it is trained to map to 3048 senones. The Whisper-based acoustic model is trained to map to 50257 tokens. To improve the robustness, before the audio feature extraction, the raw speech signals of the CGN corpus are augmented with speed perturbation with ratios 0.9, 1.0, and 1.1 [62]. After MEL filter bank feature extraction, we further apply time warping, frequency masking, and time masking for data augmentation on the log-Mel spectrogram [63].

In the second stage, we fine-tune the initial acoustic model with the dysarthric Copas data. [59] partitions the Copas data into training and test sets. The training set contains recordings from 182 speakers and the test set contains recordings from 51 speakers. The initial model is fine-tuned on this training set while using the predefined test set as validation for early stopping. Note that this test set is only used for this purpose. As argued in the introduction, speaker adaption based on impairment severity is a popular approach for dysarthric speech ASR. Therefore, we organize the Copas train data into three subsets based on the IS range: "no restriction on IS range" (referred to as "Copas"), "IS above 60" (referred to as "60up"), and "IS above 70" (referred to as "70up"), and then fine-tune the initial acoustic model with these three subsets to get three fine-tuned acoustic models. These fine-tuned models will be compared to investigate whether, like for ASR, severity-adaptation is beneficial or necessary for the SLU tasks as well. To prevent the fine-tuned model from forgetting knowledge learned from the typical speech, 4.86 h of unaugmented (raw) speech from CGN are combined with each subset of Copas to conduct the joint training during fine-tuning. The combined raw speech data is augmented with the same speed perturbation as in the former stage to triple the number of training samples. Afterwards, the spectrum extracted from the raw speech data is augmented by the same spectrogram augmentation operations. These data augmentation methods have been shown to improve ASR performance for dysarthric speech [64].

2.4 Intent decoders

The *intent decoders* map the encoder output to a multi-hot representation of intent and slot values. We investigate three types of intent decoders, the NMF decoder, the multilayer capsule network decoder, and the LSTM decoder. They differ fundamentally in the way they deal with the sequential aspects of the encoded speech representation. Because dysarthric speech is irregular in timing, it is not clear which is the best fit for the data. The *capsule decoder* has the weakest timing model: it uses an attention mechanism which is invariant to permutations in its input. The NMF decoder uses a bag-of-words approach, which is in essence also invariant to order. However, it first uses a *histogram of acoustic co-occurrence (HAC)* representation which counts bigram events in the encoded input stream. It hence becomes sensitive to order and time, though the timing model is not very strong, which might be beneficial for dysarthric speech. Finally, the LSTM decoder has the strongest sequence model capabilities. Notice that encoders are trained with ASR targets in tandem with a decoder with sequential modeling capabilities. We will therefore evaluate the combination of all three decoders with encoders and additionally discuss the performance gain by involving dysarthric speech in pre-training.

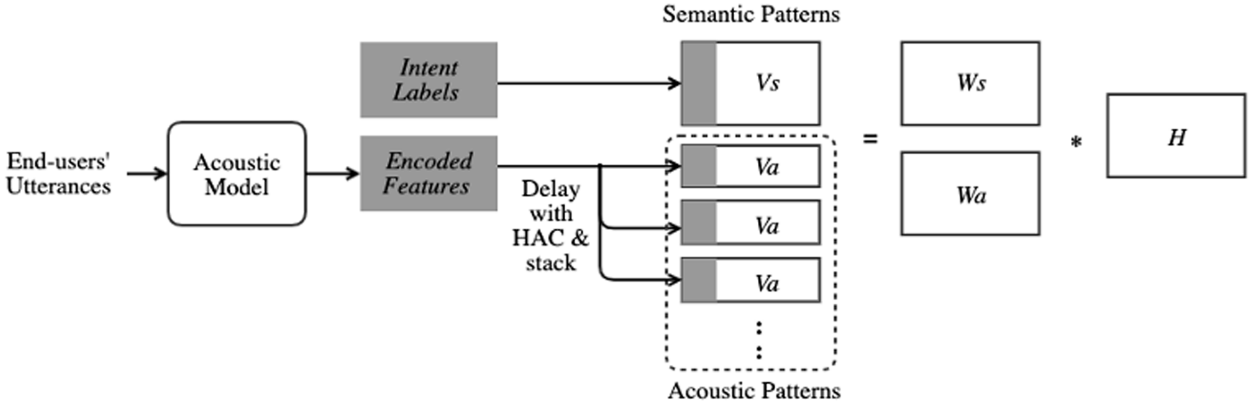


Fig. 6 The structure of the NMF-based intent decoder

2.4.1 NMF-based decoder

A key advantage of the NMF-based decoder is its low computational requirements in training. While the encoder is trained off-line on application-independent data, the decoder is trained on application-specific data provided by the user. The use of Bayesian methods for model updates and order selection in this SLU context are well-documented ([65, 66]) and feasible on low-cost hardware. The main idea of NMF is approximately decomposing a nonnegative data matrix V into two low-rank nonnegative matrixes W and H:

$$V \approx WH \tag{3}$$

where each column of V encodes an utterance plus demonstration, W is a dictionary whose columns model recurrent patterns in the data and H is an activation matrix revealing which dictionary elements occur in each utterance. The factors W and H are found by minimizing the generalized Kullback-Leibler divergence (KLD) ([67, 68]) between V and WH.

For the speech-to-intent application, as shown in Fig. 6, the model will first learn the dictionary matrix during the training process. Mathematically, the above equation can be specified as:

$$\begin{bmatrix} V_s^{(train)} \\ V_a^{(train)} \end{bmatrix} \approx \begin{bmatrix} W_s \\ W_a \end{bmatrix} H^{(train)} \tag{4}$$

where the left side of the equation is the utterance-intent training pair which is composed of two parts: the top part V_s^{train} is the semantic part which contains a binary many-hot encoding of the intent as demonstrated by the users. Each column of the bottom part V_a^{train} encodes the acoustics of a full utterance as a HAC, i.e., the bigram frequency of events at multiple delays [69]. This is a fixed-sized sentence embedding that is sensitive to the order of the acoustic events. HAC is essential here for timing encoding since NMF is a linear bag-of-words model that cannot capture order. The short delays (tens of milliseconds) in HAC make the model sensitive to order of within-word acoustic events, i.e., phone order. The longer delays (hundreds of milliseconds) make the model—to some extent—sensitive to word order. W_s and W_a on the right side of the equation are the corresponding semantic and acoustic parts in the dictionary.

During the testing process, only the acoustic part of the test sample V_a^{test} is available. The activation matrix of the test sample is found by decomposing V_a^{test} with the W_a from the training procedure:

$$V_a^{test} \approx W_a H^{test} \tag{5}$$

The estimated label of the test sample is finally obtained from:

$$V_s^{test} \approx W_s H^{test} \tag{6}$$

The actual intent is the one with a multi-hot encoding that is closest to V_s^{test} in generalized KLD.

In our application, the input sequences of the NMF decoder are generated by the encoder. A GMM model with 100 components and diagonal covariance is first applied to find the posterior probabilities of acoustic events (modeled by a single Gaussian in the mixture) encoded in the input sequences. For the HAC computation, bigram delays of 2-5-9-20 frames are used.

2.4.2 Capsule network-based decoder

A capsule network is designed to model hierarchical data [16]: here, phones form words (related to slot values) which form sentences. Capsule networks are also reported to work well on scarce training data [70]. Deeper structures are difficult to train [71], so we use just two layers. A capsule network is composed of capsule units: neural networks which output an activation

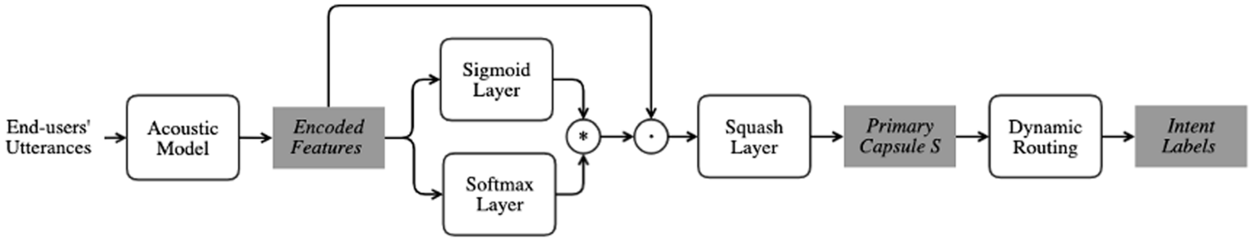


Fig. 7 The structure of the capsule network-based intent decoder

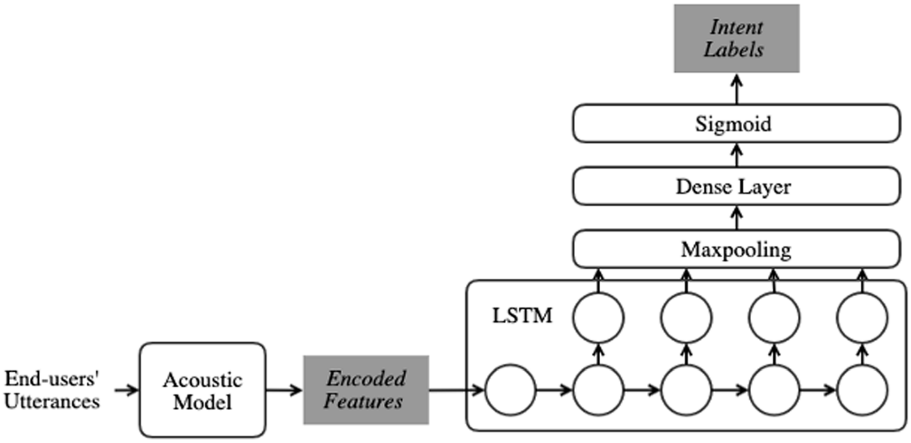


Fig. 8 The structure of the LSTM-based intent decoder

vector with length between 0 and 1 which represents the activation of the capsule, while the vector's orientation contains the latent information of the capsule. Capsules in the lower layer vote on the orientation of the capsules in the layer above them. The output capsule will be activated if a group of lower capsules agrees on its orientation. In our work, this agreement is mediated by dynamic routing. A more extensive description of the use of capsule networks in this context can be found in [14].

In our implementation, referring to Fig. 7, encoded features are taken as the inputs of a 2-layer capsule network. The encoded features F are converted to the primary capsule vectors S_i in the first capsule layer by a temporal attention α_t (*sigmoid* layer) and distributor δ_{ti} (*softmax* layer) mechanism:

$$S_i = Squash(w_s \cdot \sum_t \alpha_t \delta_{ti} F_t) \tag{7}$$

$$Squash(x) = \frac{\|x\|^2}{1 + \|x\|^2} \frac{x}{\|x\|} \tag{8}$$

Here, S_i is the representation for capsule i. There are 32 hidden capsules with 64 dimensions in the primary capsule layer. The squash function is the soft normalization in a capsule network that ensures the length of S_i lies between 0 and 1. w_s are trainable weights of the squash layer. α_t is the attention weight for each time step, which is used to filter out the unimportant time frames in the sequence (e.g., silence). δ_{ti} are the distribution weights of the distributor to assign each time step t to the hidden capsule i. α_t and δ_{ti} are calculated from:

$$\alpha_t = sigmoid(w_a \cdot F_t + b_a) \tag{9}$$

$$\delta_{ti} = softmax(w_d \cdot F_t + b_d) \tag{10}$$

here, w_a and b_a, w_d, and b_d are weights and biases of the *sigmoid* and *softmax* layers respectively.

The second capsule layer maps S_i to the intent and slot value labels via dynamic routing. Essentially, the second layer learns which acoustic evidence that triggered the first layer can be pieced together as evidence for an intent or slot value. There is one output capsule for each output label with 16 dimensions in this layer.

2.4.3 LSTM-based decoder

An RNN is a powerful neural network for sequential modeling and has shown its efficiency in SLU tasks [20, 72]. A long short-term memory (LSTM) network

Table 2 Intent representations of Domotica samples

Intent	Slot	Slot value	# of values	Command
Triple numeric values	⟨object⟩	Standing light	2	Floor lamp at 1
	⟨action⟩	Position 1	3	
Light switch	⟨light⟩	Kitchen light	6	Light in the kitchen on
	⟨state⟩	On	2	
Door control	⟨door⟩	Front door	6	Front door open
	⟨state⟩	Open	2	
Set temperature	{}	{}	1	Thermostat heating at 21

is an RNN that trains rather easily while learning long-term dependencies. In our implementation, a 1-layer 256-dimensional LSTM is applied to decode features from the acoustic model as shown in Fig. 8. The outputs of the LSTM layer are aggregated by a max-pooling layer along time to get a single vector for the entire utterance. This vector is converted to intent label probabilities with a 1024-dimensional dense layer with a sigmoid function.

3 Experiments

We will test the two mono-lingual acoustic models that are pre-trained on four different Dutch (typical and/or dysarthric) corpora from scratch and the two cross-lingual models, which already contain the knowledge gained from either ASR or SSL of multiple languages, combined with three SLU decoders. The compared systems are named according to the format "acoustic model - decoder - pre-training data," for example, "Transformer-Capsule-CGN" refers to the SLU system consisting of the transformer-based acoustic model pre-trained on the CGN corpus and using the capsule network decoder; "TDNN-LSTM-60up" refers to the SLU system consisting of the TDNN-based acoustic model fine-tuned on segments "IS above 60" of the Copas corpus and using the LSTM decoder. Considering that the two cross-lingual models have hundreds of millions (M) parameters while the former acoustic models are below 30M parameters, we will only fine-tune the cross-lingual models on the full Copas corpus and compare their overall performance.

The systems will be verified on four aspects:

1) Which pre-training strategy (supervised ASR training or SSL representation learning) works better for the downstream Dutch dysarthric speech SLU task;

2) Whether lower WER on dysarthric ASR leads to higher dysarthric SLU accuracy;

3) Whether including dysarthric speech in the acoustic model's pre-training boosts dysarthric SLU accuracy and makes it robust to variability induced by speaker and hence by pathology;

4) The necessity for designing a dysarthria-severity-dependent SLU system, i.e., pre-training acoustic models with dysarthric speech collected from selected persons with an impairment severity within a given range.

3.1 Task-specific dataset

The speech used for the SLU task is extracted from the Domotica database ([9, 73]), collected for the domain of home automation for disordered speech. The semantic information expressed in each spoken utterance is encoded into four intents including: "Triple numeric values" which describes spoken commands that set a domestic item into one of its three ranges such as the brightness level of a lamp, "light switch" describes turning on/off lights in different rooms, "door control" indicates opening/closing different doors, and "increase heat" which is used to change the temperature of the heating. These four intents are represented by 22 slot values (22 multi-hot labels) which in total form 27 occurring combinations, called command types. Table 2 shows an example of each intent as well as the number of values of each slot. This database contains 4147 utterances by 17 speakers using 38 different words, in which 36 words occur in CGN and 15 words occur in Copas. The data is collected from two recording phases (domotica3 and domotica4). Speakers with the same ID across databases are the same but recorded months later in a longitudinal study. The IS of the speakers is estimated using an automated tool built based on Copas [61] and is listed in Table 3. For seven speakers, the IS was not available at the time the

Table 3 Statistics of the Domotica database

Speaker IDs	IS	Speaker IDs	IS
41	64	28	73
32	65	29	74
33	66	34	76
30	69	40	86
35	72	17	89

research was performed and they are excluded from the experiments.

3.2 Experimental setup

The utterances for end-to-end SLU training are provided by the end-user only. The system is therefore expected to be highly efficient in terms of task-specific training data requirements. It should perform well with as few training samples as possible to minimize the users' effort. One important evaluation criterion is hence the number of training samples required for a given accuracy under the speaker-dependent setting. This property is measured in multiple ways.

We first simulate the insufficient training data setting. As demonstrated in Section 3.1, 27 command types occur in the Domotica database. Most of the speakers listed in Table 3 record all command types, except speaker 32 who records 26 types and speaker 33 who records 10 types. For each speaker, we simulate the insufficient training data scenario by randomly selecting two samples from each command type to form the training set (to ensure all command types are seen for training); the remaining samples serve as the test set. The size of the training set for each speaker therefore varies from 20 to 54 utterances, which is around 15% of the full data size. To ensure the reliability of the results, we conduct 30 experiments for each speaker with different random training and test sets. To avoid the comparison results being affected by the correlation from the arbitrary choices of the training sets, we follow the instruction of [74] to further estimate the generalization performance of each model. The accuracy metric of the SLU task is the micro-averaged F1-score for the slot values. We hence view every command as a collection of slot values that need to be detected. The metric will hence account for slots that are missed or falsely detected as well as slots that get assigned the wrong value.

Secondly, we compare the learning curves which record the accuracy as a function of the number of training samples. The abscissa of the learning curve is the total number of training utterances used for all command types jointly. To obtain robust results, cross-validation is applied. Per speaker in Domotica, the utterances are divided into 15 blocks of almost the same number of samples. The blocks are built by minimizing the inter-block Jensen-Shannon divergence of the label distribution to maximize the semantic similarity of blocks. The experiment is carried out by placing an increasing number of blocks in the training set and the rest in the test set. For each number of training blocks, 5 experiments are conducted with a different random training set. The resulting learning curves are presented by locally weighted scatterplot smoothing (LOWESS) [75].

Table 4 (Means and STD of) WER in % of the each impairment severity group

Model	Severe	Moderate	Mild	Mean	STD
TDNN	51.54	31.26	39.40	40.73	10.21
Transformer	56.83	61.69	43.22	53.91	9.57
Whisper	53.81	40.40	37.51	43.91	8.70
XLSR-53	59.25	46.09	43.54	49.63	8.43

4 Pre-training for dysarthric speech

4.1 Acoustic analysis: ASR inference

To better learn the pros and cons of each pre-training strategy for dysarthric speech representation modeling, and how the dysarthric speech representations contribute to the semantic SLU task, we first show the WER results of each pre-trained acoustic speech encoder for ASR inference on the Domotica data. All pre-trained models are initialized with typical CGN speech and further updated with the dysarthric Copas data. Therefore, the ASR inference task follows the zero-shot learning setting to explore the possible generalization from task-agnostic speakers to task-specific speakers.

Since all the transformer-based models (including Whisper and XLSR-53) are trained with character or token targets, they are evaluated without language model with a beam search CTC decoder (beam width 20). For the TDNN model, a 5-gram language model trained on the N-Best database [76] (broadcast news and spontaneous telephone speech) yielded better accuracy and therefore.

Similar to the impairment severity levels assigned to the Copas corpus, we partition the speakers of the Domotica database into three IS ranges (severity levels): speakers 30, 32, 33, and 41 with IS below 70 (severely impaired), speakers 28, 29, 34, and 35 with IS between 70 and 85 (moderately impaired), and speakers 17 and 40 with IS above 85 (mildly impaired). Mean WER results for each severity group are shown in Table 4 for each model. The standard deviation of the mean WER over the 10 speakers is also given per model.

In general, the TDNN acoustic model with the language model gets the best performance but is less robust to deviations introduced by speakers and pathology. The lower WER can also be attributed to the language model, which is absent in the other systems. Among the transformer-based acoustic models, the large cross-lingual models (Whisper and XLSR-53) show better generalization to different severity groups. Supervised ASR learning (Whisper) outperforms SSL representation learning with the fine-tuned ASR task. Comparing supervised ASR training, cross-lingual with multitask learning (Whisper) works better.

Table 5 Average SLU F1 scores per severity group. Best accuracy per severity group shown by ‡ and best SLU decoder per pre-trained encoder shown by †

SLU system	Severe	Moderate	Mild	Mean	STD
TDNN-Capsule	89.25	94.74	98.29‡	94.09	4.55
TDNN-LSTM	91.48†	96.78†	94.75	94.34†	2.67
TDNN-NMF	87.48	93.98	91.36	90.94	3.27
Transformer-Capsule	75.69	89.51	86.30	83.83	7.23
Transformer-LSTM	93.88‡	97.81‡	94.56†	95.42†	2.10
Transformer-NMF	84.99	91.90	88.80	88.56	3.46
Whisper-Capsule	88.33	96.65	96.38	93.79	7.23
Whisper-LSTM	88.51†	97.66†	96.95†	94.37†	2.10
Whisper-NMF	64.33	62.66	62.86	63.28	3.46
XLSR-53-Capsule	61.39	82.82†	84.53	76.25	12.89
XLSR-53-LSTM	65.16†	81.31	88.81†	78.43†	12.09
XLSR-53-NMF	41.95	72.61	68.60	61.05	16.67

4.2 Semantic analysis: SLU intent classification

In this section, we will first investigate the optimal combination of the pre-trained encoders and the SLU decoders for the semantic task on the dysarthric Domotica data. The performance relation between the acoustic task and the semantic task will be discussed along with the conclusions from Section 4.1.

After selecting the best-performing SLU system, a statistical analysis will be applied to the system to show whether significant improvements (or degradations) are observed by including (or excluding) task-agnostic dysarthric speech in pre-training, which mainly concerns its robustness to the speaker and pathological variability.

4.2.1 Comparisons of multiple SLU systems

In this section, we will look into SLU system performance under fixed training data size. The following experiments are conducted with insufficient training data as explained in Section 3.2. The training sets have fixed size limited to 20 to 54 samples. For each speaker, the SLU model is trained 30 times with a different definition of training and test sets.

We first summarize the average micro-F1 scores per severity group in Table 5. The four encoders are initialized with typical (CGN) speech and further updated with the dysarthric Copas data with the same setting as in Section 4.1. In this table, we use ‡ to indicate the best results within each severity group (one per column) and † to indicate the best SLU decoder for each pre-trained encoder.

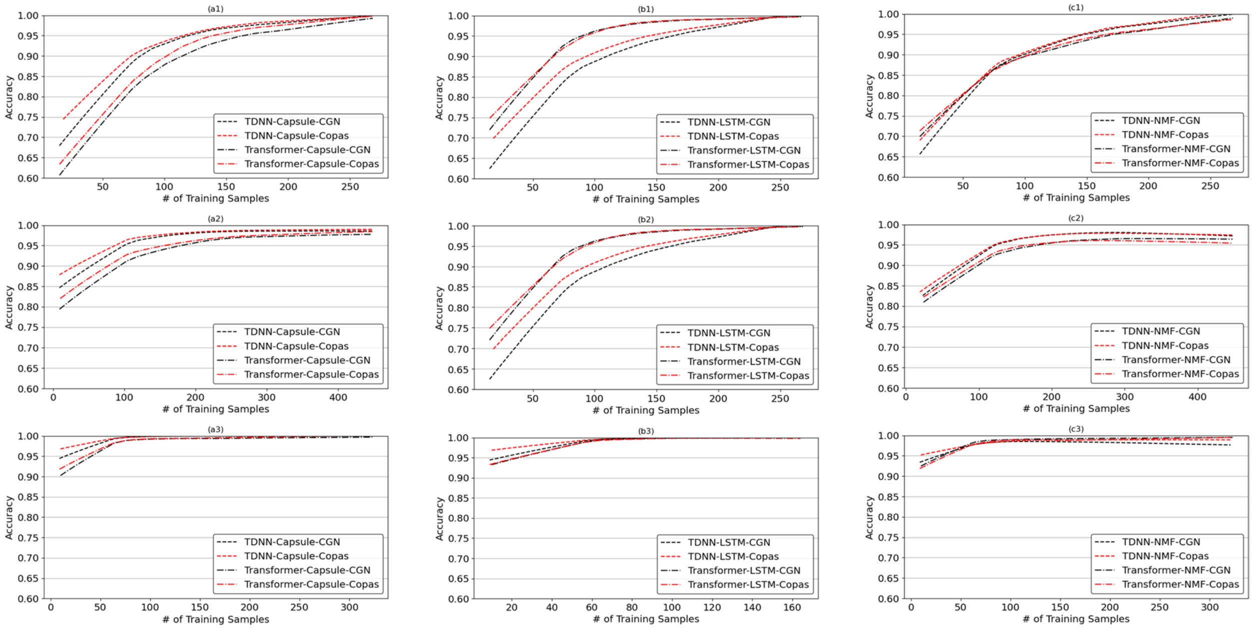
Comparing the overall F1 score of the 10 speakers with its standard deviation, the TDNN acoustic model scores best in general, which is consistent with the WERs in Table 4. Although XLSR-53 performs robustly for ASR inference, it fails to generate representations from which the decoders can easily extract semantics. As mentioned in Section 2.2.2, the Wav2Vec2.0 model does not explicitly learn a mapping from speech to semantics but learns a rich entangled representation about speech including, e.g., speaker information. Even when fine-tuning with a single-layer ASR decoder, the decoder ends up being less powerful compared to other acoustic models. The cross-lingual acoustic Whisper model achieves high accuracies with the capsule network and the LSTM decoder. However, the performance drops dramatically with the NMF decoder. Here, too, we assume the rich entangled representation (since Whisper uses multi-task learning) is too hard for the GMM layer to cluster into centroids that the decoder can map to the semantic categories.

Comparing the F1 score per severity group, the TDNN-based and transformer-based acoustic models achieve the best results by combining with either the capsule network or the LSTM decoder. However, we notice that the SLU results are not strictly related to the severity level of the impairment nor to the WERs. For example, for the transformer-based acoustic model, ASR inference on the mildly impaired group generates lower WER than on the moderately impaired group (Table 4), while the opposite is observed with all three SLU decoders (Table 5). We further investigate the possible impact of involving dysarthric speech (Copas) during pre-training of these two acoustic models.

4.2.2 Dysarthric speech SLU analysis with(out) dysarthric speech pre-training

We start with a performance comparison of the TDNN-based and transformer-based acoustic models pre-trained with and without dysarthric speech. We first show the learning curve of each model. The average learning curves over per severity group are shown in Fig. 9. The results for the three severity groups are shown with the same scale for better comparison. In each sub-figure, the black and red curves represent models pre-trained without and with the dysarthric corpus respectively. The dashed lines in each sub-figure represent the TDNN-based acoustic model while the dash-dot lines represent the transformer-based acoustic model.

Figure 9 (a1) to (a3), (b1) to (b3), and (c1) to (c3) show the average learning curves of the two acoustic models with the (a) capsule network decoder, (b) the LSTM decoder, and (c) the NMF decoder respectively. We observe significant differences between the models. Overall, within each severity group, involving the dysarthric corpus in pre-training always leads to performance gains compared to only pre-training with typical speech. For the TDNN-based acoustic model (dashed line), by



(a) (b) (c)

Fig. 9 Average learning curves for different decoders: **(a)** capsule network, **(b)** LSTM, **(c)** NMF. Rows correspond to (1) severely impaired speakers (2) moderately impaired speakers (3) mildly impaired speakers

introducing dysarthric speech, accuracies at around 20 training utterances are improved on average by 8% for severely impaired speakers (IS below 70) as shown in Fig. 9 (a1) to (c1), by 4% for moderately impaired speakers (IS ranged between 70 and 85) shown in Fig. 9 (a2) to (c2), and by only 2% for mildly impaired speakers (IS above 85) as shown in the Fig. 9 (a3) to (c3). Comparing the sub-figures vertically, i.e., from Fig. 9 (a1) to (a3), the absolute performance gains for severely impaired speakers are more important than for mildly impaired speakers when more training samples are available. The absolute improvements for mildly impaired speakers are small, but in terms of relative error rate reduction, practically relevant differences are observed, consistently with the severely and moderately impaired cases. Although the performance gains by adding the dysarthric corpus are smaller for transformer-based acoustic models than for TDNN-based models in general, the improvements of different speaker classes (severe/moderate/mild) are similar to the results of the TDNN-based model, severely impaired speakers benefit more than mildly impaired speakers.

By comparing sub-figures horizontally, i.e., from Fig. 9 (a1) to (c1), we also observe that these two acoustic models perform differently when complemented with different decoders. For the capsule network decoder, the transformer-based acoustic model performs worse than with the TDNN-based acoustic model, which is opposite of what is observed for the LSTM decoder, where the transformer-based model outperforms the TDNN-based model. For the NMF decoder, the differences between results of these two acoustic models are negligible. One significant difference of the three decoders is their capabilities in capturing dependencies within sequences. As explained in Section 2.4, the auto-regressive nature of an LSTM enables it to capture long-term dependencies and can learn how much timing can be trusted as a source of information. The NMF decoder cannot model timing as well as an LSTM, but it additionally includes order information by stacking HAC embeddings at different delays in the acoustic representations. The capsule network decoder is the weakest in processing timing. The capsule network decoder appears to perform the worst, presumably because the TDNN uses frame splicing inducing stricter temporal information than the highly flexible attention mechanism in transformers. When the later one is combined with decoder with the weakest timing

Table 6 Number of parameters in the SLU models

Model	Encoder	Intent decoder	Total
TDNN-Capsule	19*M*	737.2*K*	19.7*M*
Transformer-Capsule	27*M*	746.5*K*	27.7*M*
TDNN-LSTM	19*M*	432.7*K*	19.4*M*
Transformer-LSTM	27*M*	531.0*K*	27.5*M*

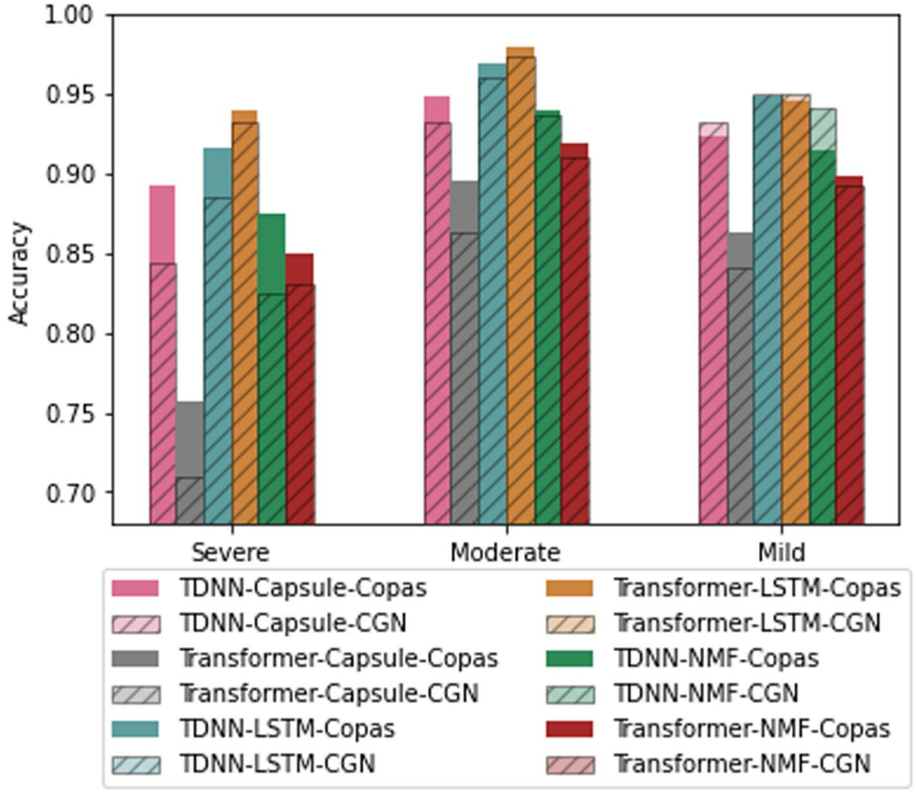


Fig. 10 Average accuracy of acoustic models fine-tuned with and without dysarthria corpus under insufficient SLU task-specific training data

modeling, a degradation is observed. Also, the training procedure of a TDNN acoustic model requires a HMM-GMM model to obtain state alignments first. This step may introduce intermediate errors on dysarthric data, due to disfluencies and acoustic mismatch. By contrast, the ASR targets for the transformer acoustic model are character-level units with end-to-end training. Therefore, the transformer acoustic model obtains better performance as shown in Fig. 9 (b1) to (b3) when the (LSTM) decoder handles the timing. The NMF decoder is able to remedy the degradation caused by insufficient timing information of the transformer model as well, but it is still not as powerful as the LSTM decoder.

Another aspect that needs to be noticed is that the improvements obtained by involving the dysarthric corpus are in general smaller for the transformer-based acoustic model than for the TDNN-based acoustic model. A cause might be the different size of these two models as recorded in Table 6: The transformer-based acoustic model has 27M parameters while the TDNN-based model has a size of 19M. Since the dysarthric corpus used for fine-tuning is small, the transformer is not likely to be as well-trained and fully adapted to the dysarthric speech leading to smaller improvements.

We have presented and discussed the models' performance averaged over multiple speakers as a function of increasing the training set size. In the next experiment, we further look into performance under fixed training data size to simulate the insufficient training data scenario. We perform statistical analysis of the differences when including (or excluding) dysarthric speech in pre-training.

The presented results are the bar plots of the average accuracy of 30 experiments and the average absolute performance differences by introducing dysarthric speech to each SLU system-pair. (By SLU system-pair, we refer to the two SLU systems which are built with the same acoustic model-decoder combination, for example TDNN-LSTM, but one is fine-tuned with dysarthric speech and the other does not contain dysarthric speech in fine-tuning.) As demonstrated by [74], evaluating two models by solely comparing the average accuracy and performance differences of the two models is not correct since the variability due to the randomness of the training set for comparisons is not taken into account. We further analyze the variance of the absolute performance differences of each SLU system-pair to test the hypothesis that such differences are significant and do not depend on the training samples we select.

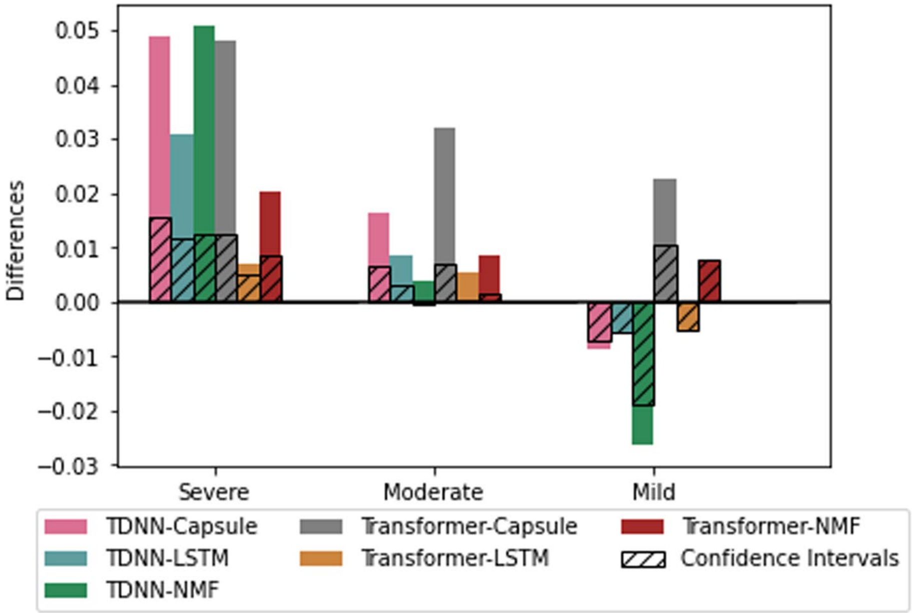


Fig. 11 Differences of acoustic models fine-tuned with and without dysarthria corpus in a setting of insufficient SLU task-specific training data

Figure 10 shows the average accuracy results per severity group. We observe that IS is indeed a factor affecting accuracy, but SLU performance is also affected by, e.g., cross-utterance consistency in speaking, word choice and grammar, resulting in a trended but non-smooth accuracy. The average accuracy of all SLU system pairs is drawn together for better comparison. The performance gains or degradations by introducing Copas data into pre-training can be read by the difference between the solid bar and the hatched bar. For example, for severely impaired speaker, introducing Copas data to the TDNN-Capsule model improves the performance by around 5% absolute. For mildly impaired speaker, the hatched bar for the TDNN-Capsule model is higher than the solid bar which indicates a performance drop. To intuitively visualize the differences of each SLU system pair, Fig. 11 shows the bar plots of the differences (the solid bar) and the corresponding confidence interval (the slash-filled bar) to evaluate the significance of the differences. The bar above the x-axis represents the performance improvement by introducing dysarthric Copas data into pre-training; otherwise, it indicates the performance degradation. The confidence intervals are calculated by the estimations of the variances of the differences using the corrected resampled t inference method [74]. A solid bar higher than the hatched bar indicates a significant difference (improvement or degradation) at 95% confidence. Hence, a significant improvement by introducing Copas data in pre-training occurs for example for the TDNN-Capsule/LSTM/NMF model for the severely impaired speaker group. A significant degradation is observed, for example for the TDNN-Capsule/NMF model for the mildly impaired speaker group. Other models with the solid bar below the hatched bar should be regarded as performing equally within our experimental setup, for example, the TDNN-LSTM and the Transformer-LSTM/NMF model of the mildly impaired group.

From Fig. 11, we observe that speakers in the severely impaired group with lower IS show larger performance gains by involving the dysarthria corpus. With increasing IS (moderate impairment), the absolute improvements become limited. In the mildly impaired group, some acoustic models fail to benefit from pre-training on the full Copas corpus. This is potentially caused by the data mismatch between testing and pre-training. As evidenced by Table 1, the full Copas corpus contains 15% utterances collected from severely impaired speakers whose speech production differs from moderately and mildly impaired or typical speakers (speakers 28, 29, 34, 35, 17, and 40 of Domotica). In general, the transformer-based acoustic model does not benefit as much from pre-training with the dysarthria corpus as the TDNN-based acoustic model. Among the three decoders, the LSTM decoder benefits the least.

Comparing results with different decoders, the capsule network decoder is worse than the LSTM decoder in general. A possible reason is the irregular timing occurring in dysarthric speech which requires a higher capability of the sequential modeling while the capsule network performs the weakest among all three decoders as discussed in Section 4.2.2; therefore, we observe the performance drops drastically when the transformer acoustic model is combined with the capsule network decoder and performs the worst among the six combinations (this combination is weakest in capturing timing information). Another concern is the different size of the two decoders. As shown in Table 6, for the TDNN-based acoustic model, the capsule network decoder has 737.2k parameters while the LSTM has 432.7k parameters. Since in this section, all models are trained with very limited data (a maximum of 54 utterances for training), the capsule network decoder is more likely to suffer from over-fitting.

From the above study on the effect of fine-tuning with the Copas data, we can conclude that including dysarthric speech in pre-training does boost the SLU tasks for dysarthric speech. However, the match between training and test data needs further investigation (Section 5). Since the transformer model seems to benefit less from pre-training on dysarthric speech than the TDNN model, we will henceforth only use the TDNN acoustic model pre-trained on the CGN corpus of typical speech as the initial model, fine-tune it on the utterances collected from the three different IS ranges (no restriction on IS range, IS above 60 and IS above 70) of the Copas data as discussed in Section 2.3.2 to further investigate whether in this setting, designing a dysarthria-severity-dependent system is beneficial for SLU tasks.

5 SLU performance with dysarthria-severity-dependent acoustic models

Section 4.2.2 shows that the performance gains of including dysarthric speech in pre-training vary with the dysarthria severity of the test speakers. We hypothesize dysarthric speakers have speech characteristics that can be grouped by intelligibility score and hence the corpus used for pre-training (full Copas) mismatches the test utterances (utterances with higher IS). We therefore build three acoustic models by fine-tuning the initial acoustic model on utterances collected from three IS ranges of the Copas corpus to create a better match. The models are referred to as “TDNN-Copas,” “TDNN-60up,” and “TDNN-70up” respectively. To avoid averages hiding individual differences, we perform the comparisons on the speaker with highest and lowest IS in the severity group.

The initial TDNN acoustic model pre-trained solely on typical speech is referred to as “TDNN-CGN.” The four acoustic models are then combined with the three SLU decoders and tested on each speaker in the Domotica test. The experiments are conducted under the same insufficient training data setting as in Section 4.2.1 repeated 30 times with different training and test sets. The training sets are formed by randomly selecting two samples of each command type. The estimations of the variances of differences [74] are applied to each SLU system-pair to test the significance of comparisons.

5.1 Performance for severely impaired speakers

We present results for the speaker 41 with IS 64.22 and speaker 30 with IS 68.99. In Fig. 12, the accuracy of 30 repeated experiments of each model are given by box plots with significance test results. We indicate significance test results at 95% confidence in the box plots using symbols $*$ (for significant) and *ns* (for “not significant”). For speaker 41, with the capsule network decoder, fine-tuning on the full Copas performs the best. The $*$ symbol between the results of “TDNN-CGN” and “TDNN-Copas” indicates that “TDNN-Copas” is significantly better than pre-training solely on CGN. Also for the NMF decoder we observe that fine-tuning on the full Copas performs best. The performances of “TDNN-60up” and “TDNN-70up” do not show significant differences. For the LSTM decoder, “TDNN-Copas” and “TDNN-60up” perform equally and better than other models.

For speaker 30, with the capsule network decoder, fine-tuning on Copas still outperforms other fine-tuning designs. For the LSTM decoder, “TDNN-Copas” and “TDNN-60up” perform equally and better than other models. For the NMF decoder, the differences between “TDNN-CGN,” “TDNN-Copas,” and “TDNN-70up” are not significant while “TDNN-60up” significantly outperforms other models.

Therefore, for severely impaired speakers with IS below 70, “TDNN-Copas” or “TDNN-60up” is the best design.

5.2 Performance for moderately impaired speakers

For moderately impaired speakers, we show results of the speaker 35 with an IS of 72.33 and speaker 34 with IS 76.2 in Fig. 13.

For speaker 35, with the three decoders, fine-tuning on Copas is significantly better than pre-training on CGN. For the capsule network, fine-tuning on “TDNN-60up” and “TDNN-70up” perform equally and better than “TDNN-Copas.” For the LSTM and the NMF decoder, “TDNN-70up” is the most suitable choice.

For speaker 34, with the capsule network and LSTM decoders, “TDNN-Copas” is significantly better than “TDNN-CGN.” With the NMF decoder, these benefits are not obvious. But for all three decoders, “TDNN-60up” significantly outperforms other models. Hence,

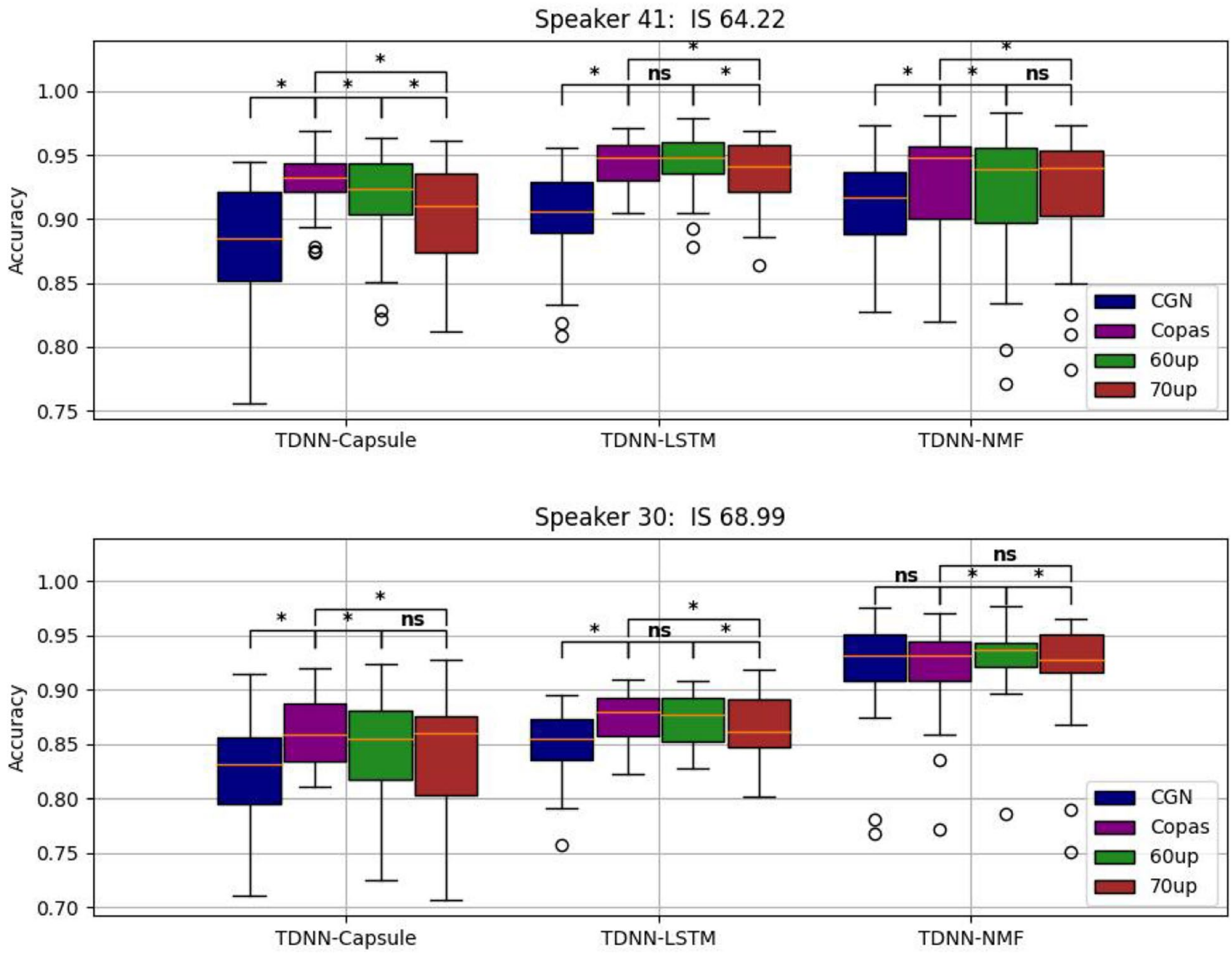


Fig. 12 Performance of the TDNN acoustic models combined with the capsule network, LSTM, and NMF decoder for SLU with insufficient task-specific training data for the speaker 41 and the speaker 30

for speakers with IS above 70, the models cannot benefit from training on the full Copas data. Excluding the utterances with IS below 60 from fine-tuning is always better.

Therefore, for moderately impaired speakers, with IS in the range 70 to 85, "TDNN-60up" or "TDNN-70up" is the best design.

5.3 Performance for mildly impaired speakers

For mildly impaired speakers, we show results of speaker 40 with an IS of 85.5 and speaker 17 with IS of 88.57 in Fig. 14.

For speaker 40, with the capsule network and NMF decoders, "TDNN-CGN" is significantly better than "TDNN-Copas." These degradations are caused by involving poorly articulated utterances in the pre-training while evaluating the system on high IS speakers. For the LSTM decoder, such degradation is not significant. In general, "TDNN-70up" is the best though only minor effects of pre-training are observed for high IS speakers for all three decoders.

For speaker 17, with the capsule network and LSTM decoder, fine-tuning on dysarthric speech still helps. "TDNN-70up" is in general the best, but for the NMF decoder, it is hard to tell which data set is the best choice.

In summary, for mildly impaired speakers with an IS above 85, pre-training on severely dysarthric speech can easily cause degradation, while "TDNN-70up" is the best compromise.

From the above comparisons between fine-tuned models, we conclude that it is wise to adapt the models with speech of similar impairment severity levels to achieve a higher performance gain. However, comparing the improvement obtained by pre-training on similarly impaired speakers, gains are limited for SLU. Considering the great efforts of collecting speech from severely impaired persons as well as the high risks of possible degradation, designing a dysarthria-severity-dependent

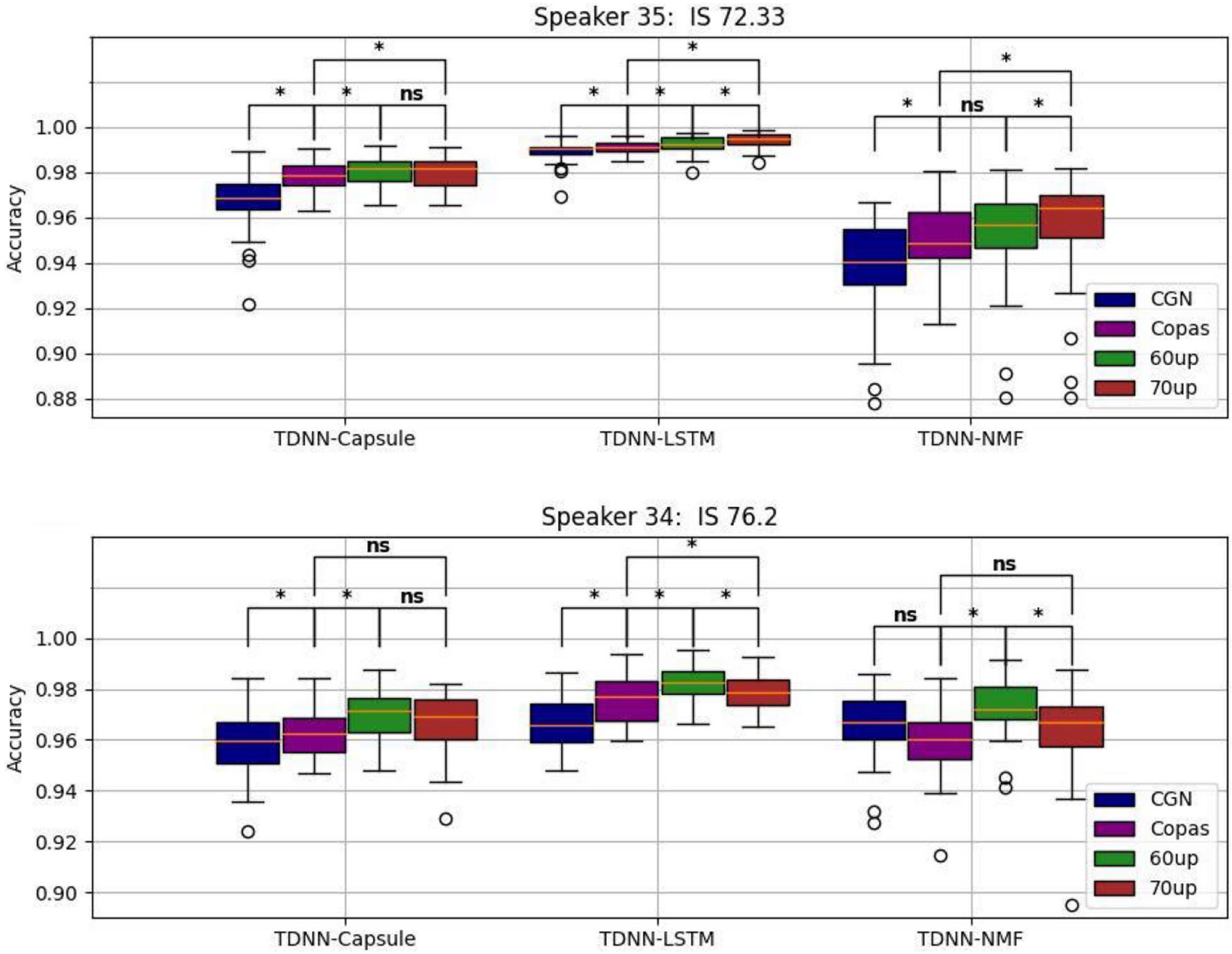


Fig. 13 Performance of the TDNN acoustic models combined with the capsule network, LSTM, and NMF decoder for SLU with insufficient task-specific training data for the speaker 35 and the speaker 34

system by pre-training acoustic models with dysarthric speech collected from selected impairment severity is not necessary for dysarthric speech SLU tasks, unlike what is observed for ASR tasks. The most beneficial choice is to fine-tune acoustic models with utterances from mildly impaired speakers, which on the one hand contain general dysarthria characteristics for modeling and leads to acceptable improvements in most cases and, on the other hand, does not possess strong deviations in pronunciation and timing and therefore will not cause any degradation as observed when fine-tuning with speech from all impairment severities.

6 Conclusions

In this work, we design end-to-end SLU systems for dysarthric speech and investigate to which extent the dysarthric SLU task benefits from pre-training with ASR targets on dysarthric speech. Two pre-training strategies, i.e., supervised ASR target and the SSL representation learning, are compared in this work. Though our evaluation is restricted to Dutch dysarthric speech, the methodology is not language-specific, which suggests our findings extend to other languages. Similar limitations hold for dysarthria type and SLU domain.

The designed SLU system consists of a pre-trained speech representations encoder and a SLU decoder to map the encoded features to the intent slots. We present four types of acoustic models for this task: a *supervised* mono-lingual TDNN/TDNN-F acoustic model which is trained with ASR targets, a *supervised* mono-lingual transformer acoustic model which is trained with ASR targets, a *supervised* cross-lingual transformer acoustic model (Whisper) which is trained with multiple tasks, and a *self-supervised* cross-lingual Wav2Vec2.0 model (XLSR-53) which is trained with masked prediction. The acoustic model is trained and fine-tuned in two phases. We first construct an initial acoustic model trained or continually trained on a large

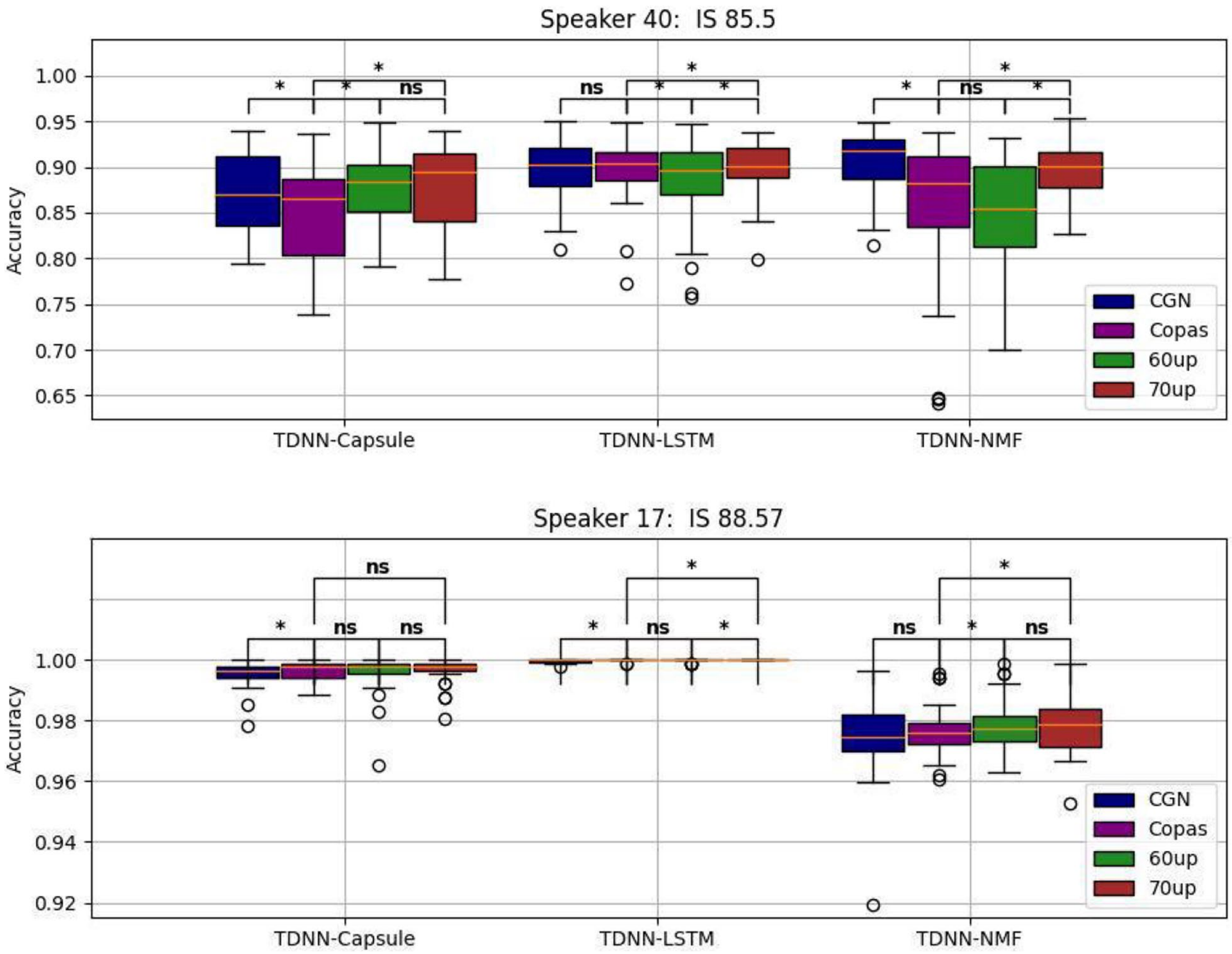


Fig. 14 Performance of the TDNN acoustic models combined with the capsule network, LSTM, and NMF decoder for SLU with insufficient task-specific training data for the speaker 40 and the speaker 17

corpus of typical Dutch speech and then fine-tune this model with a mixture of dysarthric and typical speech to model the distributions of dysarthric speech. The four acoustic models are combined with three types of successful SLU decoders for semantic inference: the NMF decoder, the multilayer capsule network decoder, and the recurrent neural network LSTM decoder.

The designed SLU systems are evaluated on a public Dutch dysarthric dataset. Among these acoustic models, both from the perspective of acoustic and semantic analysis, the supervised mono-lingual acoustic model is still optimal for the semantic SLU tasks, although the SSL and cross-lingual acoustic models exhibit more robust ASR inference. By introducing the IS for each speaker to evaluate the impairment severity and comparing the performances of acoustic models pre-trained on utterances belonging to different severity levels, we conclude that dysarthric end-to-end SLU systems can significantly benefit from knowledge transfer through pre-training on dysarthric speech with ASR targets, although it is wise to adapt the models with speech of similar impairment severity levels to maximize the performance gains or avoid degradation. Considering the obstacles in collecting severely/moderately impaired speech, pre-training with data sourced from mildly impaired speakers is the most beneficial choice for dysarthric speech SLU tasks in general, unlike for the ASR task, where strict dysarthria-severity-dependent acoustic models need to be applied. As dysarthric speech shows larger deviations in timing, a strong capability to process timing information is important for E2E dysarthric SLU systems. Among different combinations of the acoustic models with the three decoders, the LSTM decoder shows the best SLU accuracy for all four encoders and shows the least variation over speakers.

Abbreviations

ASR	Automatic speech recognition
BNF	Bottleneck features
CGN	Corpus Gesproken Nederlands
CMVN	Cepstral mean and variance normalization
CTC	Connectionist temporal classification
DIA	Dutch intelligibility assessment
E2E	End-to-end
IS	Intelligibility score
HAC	Histogram of acoustic co-occurrence
KLD	Kullback-Leibler divergence
LOWESS	Locally weighted scatterplot smoothing
LSTM	Long short-term memory
NLU	Natural language understanding
NMF	Non-negative matrix factorization
SLU	Spoken language understanding
SSL	Self-supervised learning
TDNN	Time-delay neural network
UA	Universal Access
WER	Word error rate

Acknowledgements
Not applicable.

Authors' contributions
Pu Wang conducted the research and performed the experiments. Hugo Van hamme guided and supervised the project. All authors read and approved the final manuscript.

Funding
The research was supported by the program of China Scholarship Council No.201906090275, the Flemish Government under "Onderzoeksprogramma AI Vlaanderen," and FWO-SBO grant S004923N: NELF.

Availability of data and materials
The data sets analyzed during the current study are available from CGN: http://lands.let.ru.nl/cgn/, Copas: https://taalmaterialen.ivdnt.org/download/tstc-corpus-pathologische-en-normale-spraak-copas/, and Domotica: https://www.esat.kuleuven.be/psi/spraak/downloads/.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 19 October 2022 Accepted: 14 March 2023
Published online: 07 April 2023

References

1. M. Jefferson, *in Retrieved from the University of Minnesota Digital Conservancy*, Usability of automatic speech recognition systems for individuals with speech disorders: past, present, future, and a proposed model (2019)
2. F. Ballati, F. Corno, L. De Russis, in *Intelligent Environments 2018*, "Hey Siri, do you understand me?": Virtual assistants and dysarthria. Rome, Italy: IOS Press (2018), pp. 557–566
3. E. Bastianelli, G. Castellucci, D. Croce, R. Basili, D. Nardi, Structured learning for spoken language understanding in human-robot interaction. Int. J. Robot. Res. **36**(5–7), 660–683 (2017)
4. D. Woszczyk, S. Petridis, D. Millard, in *Interspeech 2020*, Domain adversarial neural networks for dysarthric speech recognition (International Speech Communication Association (ISCA), 2020), pp. 3875–3879
5. Y. Takashima, T. Takiguchi, Y. Ariki, in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, End-to-end dysarthric speech recognition using multiple databases. Brighton, United Kingdom: IEEE pp. 6395–6399
6. L. Wu, D. Zong, S. Sun, J. Zhao, in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, A sequential contrastive learning framework for robust dysarthric speech recognition. Toronto, Ontario, Canada: IEEE pp. 7303–7307
7. J.P. Bigham, R. Kushalnagar, T.H.K. Huang, J.P. Flores, S. Savage, in *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*, On how deaf people might use speech to control devices. Pittsburgh, PA, USA: ACM(2017), pp. 383–384
8. V. Renkens, ASSIST: Assistive speech interface for smart technologies. Ph.D. thesis, KU Leuven, Department of Electrical Engineering-ESAT (2019)
9. B. Ons, J.F. Gemmeke, H. Van hamme, The self-taught vocal interface. EURASIP J. Audio Speech Music Process. **2014**(1), 1–16 (2014)
10. L. Lugosch, M. Ravanelli, P. Ignoto, V.S. Tomar, Y. Bengio, in *Interspeech 2019*, Speech model pre-training for end-to-end spoken language understanding (International Speech Communication Association (ISCA))
11. H. Christensen, S. Cunningham, C. Fox, P. Green, T. Hain, in *Interspeech 2012*, A comparative study of adaptive, automatic recognition of disordered speech (International Speech Communication Association (ISCA))
12. J.F. Gemmeke, S. Sehgal, S. Cunningham, H. Van hamme, in *2014 IEEE Spoken Language Technology Workshop (SLT)*, Dysarthric vocal interfaces with minimal training data. South Lake Tahoe, NV, USA: IEEE pp. 248–253
13. V. Renkens, H. Van hamme, in *Interspeech 2018*, Capsule networks for low resource spoken language understanding (International Speech Communication Association (ISCA)), pp. 601–605
14. J. Poncelet, H. Van hamme, in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Multitask learning with capsule networks for speech-to-intent applications. Changed to Virtual Conference: IEEE pp. 8494–8498
15. P. Wang, H. Van hamme, in *2021 IEEE Spoken Language Technology Workshop (SLT)*, A light transformer for speech-to-intent applications. Changed to Virtual Conference: IEEE pp. 997–1003
16. S. Sabour, N. Frosst, G.E. Hinton, in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, Dynamic routing between capsules. Long Beach, CA, USA: NIPS (2017), pp. 3859–3869
17. H.W. Fentaw, T.H. Kim, Design and investigation of capsule networks for sentence classification. Appl. Sci. **9**(11), 2200 (2019)
18. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, in *Proceedings of the 31st International Conference on Neural Information Processing Systems* (NeurIPS), Attention is all you need. Long Beach, CA, USA: NIPS (2017), pp. 5998–6008
19. P. Haghani, A. Narayanan, M. Bacchiani, G. Chuang, N. Gaur, P. Moreno, R. Prabhavalkar, Z. Qu, A. Waters, in *2018 IEEE Spoken Language Technology Workshop (SLT)*, From audio to semantics: Approaches to end-to-end spoken language understanding. Athens, Greece: IEEE pp.720–726
20. D. Serdyuk, Y. Wang, C. Fuegen, A. Kumar, B. Liu, Y. Bengio, in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Towards end-to-end spoken language understanding. Calgary, Alberta, Canada: IEEE pp. 5754–5758
21. Y.P. Chen, R. Price, S. Bangalore, in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Spoken language understanding without speech recognition. Calgary, Alberta, Canada: IEEE pp. 6189–6193
22. N. Tomashenko, A. Caubrière, Y. Estève, in *Interspeech 2019*, Investigating adaptation and transfer learning for end-to-end spoken language understanding from speech (International Speech Communication Association (ISCA)), pp. 824–828
23. R. Price, in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, End-to-end spoken language understanding without matched language speech model pretraining data. Barcelona, Spain: IEEE pp. 7979–7983
24. P. Wang, H. Van hamme, Pre-training for low resource speech-to-intent applications. arXiv preprint arXiv:2103.16674 (2021)
25. S. Bhosale, I. Sheikh, S.H. Dumpala, S.K. Kopparapu, in *Interspeech 2019*, End-to-end spoken language understanding: Bootstrapping in low resource scenarios (International Speech Communication Association (ISCA)), pp. 1188–1192
26. A. Baevski, H. Zhou, A. Mohamed, M. Auli, in *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS)*, Wav2Vec 2.0: a framework for self-supervised learning of speech representations. Changed to Virtual Conference: NIPS (2020)

27. W.N. Hsu, B. Bolte, Y.H. Tsai, K. Lakhotia, R. Salakhutdinov, A. Mohamed, Hubert: Self-supervised speech representation learning by masked prediction of hidden units. IEEE/ACM Trans Audio Speech Lang. Process. **10**(3) (2021). https://doi.org/10.1109/TASLP.2021.3122291
28. S. Pascual, M. Ravanelli, J. Serra, A. Bonafonte, Y. Bengio, in *Interspeech 2022*, Learning problem-agnostic speech representations from multiple self-supervised tasks (International Speech Communication Association (ISCA)), pp. 161–165
29. A.T. Liu, S.w. Yang, P.H. Chi, P.c. Hsu, H.y. Lee, in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mockingjay: unsupervised speech representation learning with deep bidirectional transformer encoders. Barcelona, Spain: IEEE pp. 6419–6423
30. A. Baevski, W.N. Hsu, A. Conneau, M. Auli, in *Proceedings of the 35th International Conference on Neural Information Processing Systems (NeurIPS)*, Unsupervised speech recognition (2021), pp. 27826–27839
31. A. Hernandez, P.A. Pérez-Toro, E. Nöth, J.R. Orozco-Arroyave, A. Maier, S.H. Yang, in *Interspeech 2022*, Cross-lingual self-supervised speech representations for improved dysarthric speech recognition (International Speech Communication Association (ISCA)), pp. 51–55
32. Y. Peng, S. Arora, Y. Higuchi, Y. Ueda, S. Kumar, K. Ganesan, S. Dalmia, X. Chang, S. Watanabe, in *2022 IEEE Spoken Language Technology Workshop (SLT)*. A study on the integration of pre-trained ssl, asr, lm and slu models for spoken language understanding, pp. 406–413
33. Z. Yue, H. Christensen, J. Barker, in *Interspeech 2020*, Autoencoder bottleneck features with multi-task optimisation for improved continuous dysarthric speech recognition (International Speech Communication Association (ISCA))
34. E. Yılmaz, V. Mitra, G. Sivaraman, H. Franco, Articulatory and bottleneck features for speaker-independent asr of dysarthric speech. Comput. Speech Lang. **58**, 319–334 (2019)
35. E. Hermann, M.M. Doss, in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Dysarthric speech recognition with lattice-free mmi. Barcelona, Spain: IEEE pp. 6109–6113
36. P. Wang, B. BabaAli, H. Van hamme, in *Interspeech 2021*, A study into pre-training strategies for spoken language understanding on dysarthric speech (International Speech Communication Association (ISCA)), pp. 36–40
37. J. Devlin, M.W. Chang, K. Lee, K. Toutanova, in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (NAACL)*, BERT: Pre-training of deep bidirectional transformers for language understanding (Association for Computational Linguistics (ACL), 2019), pp. 4171–4186
38. Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, in *2020 International Conference on Learning Representations (ICLR)*, ALBERT: a lite bert for self-supervised learning of language representations. Changed to Virtual Conference: ICLR
39. Y. Sun, S. Wang, Y. Li, S. Feng, H. Tian, H. Wu, H. Wang, in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34. ERNIE 2.0: a continual pre-training framework for language understanding. New York, NY, USA: IJCAI (2020), pp. 8968–8975
40. Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R.R. Salakhutdinov, Q.V. Le, in *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS)*, XLNet: Generalized autoregressive pretraining for language understanding (2019), 5753-5763
41. A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P. von Platen, Y. Saraf, J. Pino, A. Baevski, A. Conneau, M. Auli, in *Interspeech 2022*, XLSR: Self-supervised cross-lingual speech representation learning at scale (International Speech Communication Association (ISCA)), pp. 2278–2282
42. B. Vachhani, C. Bhat, S.K. Kopparapu, in *Interspeech 2018*, Data augmentation using healthy speech for dysarthric speech recognition (International Speech Communication Association (ISCA)), pp. 471–475
43. J. Shor, D. Emanuel, O. Lang, O. Tuval, M. Brenner, J. Cattiau, F. Vieira, M. McNally, T. Charbonneau, M. Nollstadt, et al., in *Interspeech 2019*, Personalizing ASR for dysarthric and accented speech with limited data (International Speech Communication Association (ISCA)), pp. 784–788
44. A. Radford, J.W. Kim, T. Xu, G. Brockman, C. McLeavey, I. Sutskever, Robust speech recognition via large-scale weak supervision. arXiv preprint arXiv:2212.04356 (2022)
45. A. Conneau, A. Baevski, R. Collobert, A. Mohamed, M. Auli, in *Interspeech 2021*, Unsupervised cross-lingual representation learning for speech recognition (International Speech Communication Association (ISCA)), pp. 2426–2430
46. M.J. Kim, J. Yoo, H. Kim, in *Interspeech 2013*, Dysarthric speech recognition using dysarthria-severity-dependent and speaker-adaptive models. (International Speech Communication Association (ISCA)), pp. 3622–3626
47. F. Xiong, J. Barker, Z. Yue, H. Christensen, in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Source domain data selection for improved transfer learning targeting dysarthric speech recognition. Barcelona, Spain: IEEE pp. 7424–7428
48. M.B. Mustafa, S.S. Salim, N. Mohamed, B. Al-Qatab, C.E. Siong, Severity-based adaptation with limited data for asr to aid dysarthric speakers. PloS ONE **9**(1), 86285 (2014)
49. Y. Zhao, C. Ni, C.C. Leung, S.R. Joty, E.S. Chng, B. Ma, in *Interspeech 2020*, Speech transformer with speaker aware persistent memory (International Speech Communication Association (ISCA)), pp. 1261–1265
50. S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N.E.Y. Soplin, J. Heymann, M. Wiesner, N. Chen, et al., in *Interspeech 2018*, ESPnet: end-to-end speech processing toolkit (International Speech Communication Association (ISCA)), pp. 2207–2211
51. V. Peddinti, D. Povey, S. Khudanpur, in *Interspeech 2015*, A time delay neural network architecture for efficient modeling of long temporal contexts (International Speech Communication Association (ISCA)), pp. 3214–3218
52. D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, S. Khudanpur, in *Interspeech 2018*, Semi-orthogonal low-rank matrix factorization for deep neural networks. (International Speech Communication Association (ISCA)), pp. 3743–3747
53. D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, K. Vesely, in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, The kaldi speech recognition toolkit (IEEE Signal Processing Society)
54. T. Matsushima, Dutch dysarthric speech recognition: Applying self-supervised learning to overcome the data scarcity issue. Ph.D. thesis, University of Groningen (2022)
55. H. Kim, M. Hasegawa-Johnson, A. Perlman, J. Gunderson, T.S. Huang, K. Watkin, S. Frame, in *Interspeech 2008*, Dysarthric speech database for universal access research (International Speech Communication Association (ISCA)), pp. 1741–1744
56. F. Rudzicz, A.K. Namasivayam, T. Wolff, The torgo database of acoustic and articulatory speech from speakers with dysarthria. Lang. Resour. Eval. **46**(4), 523–541 (2012)
57. X. Menendez-Pidal, J.B. Polikoff, S.M. Peters, J.E. Leonzio, H.T. Bunnell, in *Proceeding of Fourth International Conference on Spoken Language Processing (ICSLP)*, vol. 3. The nemours database of dysarthric speech (IEEE, 1996), pp. 1962–1965
58. I. Schuurman, M. Schouppe, H. Hoekstra, T. Van der Wouden, in *Proceedings of 4th International Workshop on Linguistically Interpreted Corpora (LINC-03) at EACL 2003*, CGN, an annotated corpus of spoken dutch. Budapest, Hungary: EACL (2003)
59. G. Van Nuffelen, M. De Bodt, C. Middag, J.P. Martens, *Dutch corpus of pathological and normal speech (copas)* (Antwerp University Hospital and Ghent University, Tech. Rep, 2009)
60. M. De Bodt, C. Guns, G. Van Nuffelen, S. Stevelinck, J. Van Borsel, G. Verbeke, A. Versonnen, F. Wuyts, *NSVO: Nederlandstalig SpraakVerstaanbaarheidsOnderzoek* (Vlaamse Vereniging voor Logopedisten (VVL), Belgium, 2006)
61. C. Middag, Automatic analysis of pathological speech. Ph.D. thesis, Ghent University (2012)
62. T. Ko, V. Peddinti, D. Povey, S. Khudanpur, in *Interspeech 2015*, Audio augmentation for speech recognition (International Speech Communication Association (ISCA)), pp. 3586–3589
63. D.S. Park, W. Chan, Y. Zhang, C.C. Chiu, B. Zoph, E.D. Cubuk, Q.V. Le, in *Interspeech 2019*, Specaugment: a simple data augmentation method for automatic speech recognition (International Speech Communication Association (ISCA)), pp. 2613–2617
64. C. Bhat, A. Panda, H. Strik, in *Interspeech 2022*, Improved asr performance for dysarthric speech using two-stage dataaugmentation (International Speech Communication Association (ISCA)), pp. 46–50

65. J. Driesen, H. Van hamme, Modelling vocabulary acquisition, adaptation and generalization in infants using adaptive bayesian plsa. Neurocomputing. **74**(11), 1874–1882 (2011)
66. V. Renkens, H. Van hamme, Automatic relevance determination for non-negative dictionary learning in the gamma-poisson model. Sign. Process. **132**, 121–133 (2017)
67. C. Févotte, J. Idier, Algorithms for nonnegative matrix factorization with the β-divergence. Neural Comput. **23**(9), 2421–2456 (2011)
68. E. Gaussier, C. Goutte, in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, Relation between plsa and nmf and implications. Salvador, Brazil: ACM (2005), pp. 601–602
69. H. Van hamme, in *Interspeech 2008*, Hac-models: a novel approach to continuous speech recognition (International Speech Communication Association (ISCA)), pp. 2554–2557
70. A. Jiménez-Sánchez, S. Albarqouni, D. Mateus, in *Intravascular Imaging and Computer Assisted Stenting and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis*. ed. by D. Stoyanov, Z. Taylor, S. Balocco, R. Sznitman, A. Martel, L. Maier-Hein, L. Duong, G. Zahnd, S. Demirci, S. Albarqouni, S.L. Lee, S. Moriconi, V. Cheplygina, D. Mateus, E. Trucco, E. Granger, P. Jannin. Capsule networks against medical imaging data challenges (Springer International Publishing, Cham, 2018), pp.150–160
71. D. Peer, S. Stabinger, A. Rodríguez-Sánchez, Limitation of capsule networks. Pattern Recognition Letters **144**, 68–74 (2021). https://doi.org/10.1016/j.patrec.2021.01.017
72. G. Mesnil, Y. Dauphin, K. Yao, Y. Bengio, L. Deng, D. Hakkani-Tur, X. He, L. Heck, G. Tur, D. Yu, G. Zweig, Using recurrent neural networks for slot filling in spoken language understanding. IEEE/ACM Transactions on Audio, Speech, and Language Processing **23**(3), 530–539 (2015). https://doi.org/10.1109/TASLP.2014.2383614
73. N.M. Tessema, B. Ons, J. van de Loo, J. Gemmeke, G. De Pauw, W. Daelemans, H. Van hamme, Metadata for corpora patcor and domotica-2. Technical report KUL/ESAT/PSI/1303, KU Leuven, ESAT, Leuven, Belgium (2013)
74. C. Nadeau, Y. Bengio, in *Proceedings of the 12th International Conference on Neural Information Processing Systems (NeurIPS)*, Inference for the generalization error (1999), pp. 307-313
75. W.S. Cleveland, Robust locally weighted regression and smoothing scatterplots. Journal of the American statistical association **74**(368), 829–836 (1979)
76. D.A. van Leeuwen, N-best 2008: a benchmark evaluation for large vocabulary speech recognition in Dutch. Essential Speech and Language Technology for Dutch: Results by the STEVIN programme. Springer Berlin Heidelberg (2013), pp. 271–288

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.