Università
degli Studi
di Ferrara

ISTITUTO
ITALIANO DI
TECNOLOGIA

DOCTORAL COURSE IN TRANSLATIONAL NEUROSCIENCE AND NEUROTECHNOLOGIES

CYCLE XXXIII

COORDINATOR Prof. Luciano Fadiga

Scientific/Disciplinary Sector (SDS) BIO/09

On Deep Learning strategies to address Automatic Speech Recognition (ASR) for dysarthric speech

Candidate:
Rosanna Turrisi

Supervisor:
Dr. Leonardo Badino

Year 2017/2020

Acknowledgements

I firstly would like to thank my supervisor Leonardo Badino who gave me the possibility to choose the thesis topic closest to my interests and the freedom to explore different research lines. Thank you for the patience you had in teaching me that led me from being a student to being a researcher. I would also like to thank professor Luciano Fadiga for giving me the possibility to carry out this work and the trust he put on me when the circumstances were not ideal. Then, I would like to thank professor Rémi Flamary whose enthusiasm made me love research again. I learned a lot under your supervision.

Last but not least, I need to thank Noemi Montobbio, the most present figure in my career since university. Our stimulating discussions and your advice have always guided me during these years.

A special thank to all patetients, and especially individuals affected by dysarthria, who give their time and, despite the fatigue, contribute to the research.

Contents

Chapter 1

Introduction

In this thesis we introduce deep learning based approaches to improve the performance of Automatic Speech Recognition (ASR) systems on dysarthric speech. In particular, we address this problem from three main perspectives.

The first one is inspired by neurophysiological studies and proposes the integration of speech production knowledge in the ASR system. The second one confronts a more general problem known in machine learning as domain adaptation that, roughly speaking, aims at learning from a source data distribution a well performing model on a second dataset (e.g., a different speaker dataset). The last approach attempts to build more robust and efficient system recognizers by reducing the ASR vocabulary, and focusing on Voice Command Speech Recognition.

To understand the arguments that encouraged us to investigate these three research lines, it is crucial to bear in mind which are the characteristics of dysarthric speech and the weakness of current ASR systems.

Dysarthria is one of the most common Motor Speech Disorders (MSDs), that are defined as disorders in planning, programming control, or execution of speech caused by neurological impairments. Specifically, dysarthria consists in the disruption of the normal control of the vocal tract musculature. The global incidence of this disorder in the world population is unknown, but it is commonly recorded as a consequence of widespread diseases such as cerebral palsy, Parkinson's disease, Amyotrophic Lateral Sclerosis (ALS), right hemisphere syndrome or dementia. It can also follow brain injury or stroke.
Moreover, it can present different degrees of severity (e.g., moderate, middle, severe). Clinical assessment tools, such as Frenchay Dysarthria Assessment (FDA) [1], Computerized Assessment of Intelligibility of Dysarthric Speech (CAIDS) [2], Therapy Outcome Measure (TOM) [3] provide the subject's impairment degree.
Based on the location of the neurological damage, dysarthria can also be classified into several different types (e.g., flaccid, spastic). Each type expresses different speech characteristics. However, it is possible to individuate common tendencies in dysarthric patients, such as mumbled speech, an acceleration or deceleration in the speaking rate, abnormal pitch and rhythm. Further, individuals with MSDs are more subject to a fatigue factor that affects the voice.

All this can severely and negatively impact the intelligibility of speech. Research literature suggests that increasing severity of dysarthria often correlates with decreasing degrees of speech intelligibility [4–6].
This loss of communication prevents patients from participating in many activities and may lead to social isolation, reducing the quality of life (QoL). The goal of management of dysarthria is to optimize communication effectiveness for as long as possible. Even if speech therapy can delay the progression of dysarthria, ASR based techniques are currently the only possible choice to enhance the QoL of the patients, especially in the most advanced phases of dysarthria.

For instance, such technologies may allow individuals with dysarthria to interact with devices by using the voice when an abnormal muscle activity does not permit a motor control. Or also, these may convert the dysarthric speech into a text or enhance it when it is characterized by a poor intelligibility.

Unfortunately, modern automatic speech recognition (ASR) is ineffective at understanding relatively unintelligible speech caused by dysarthria. As reported in [7–10], traditional representations in ASR such as Hidden Markov models (HMMs) trained for speaker independence that achieve 84.8% word-level accuracy for non-dysarthric speakers might achieve less than 4.5% accuracy in presence of severely dysarthric speech on short sentences. Recently, more accurate dysarthric speech recognition systems have been developed by using deep learning based approaches [11–13]. However, in case of severe disability, the ASR performance still remains poor. For instance, we conducted some preliminary experiments on a subset of the TORGO dataset [14], in collaboration with F. Rudzicz, in which we tested Google Speech API and IBM speech-to-text systems. We found that they misrecognize more than 80% of the words in single word utterances, while the human error was 30%.

Causes of poor performance may include slurred speech, weak or imprecise articulatory contacts, weak respiratory support, low volume, incoordination of the respiratory stream, hypernasality, and reduced intelligibility [6]. Additionally, dysarthric speech is not sufficiently covered in the training datasets of state-of-the-art commercial ASR systems. Indeed, only few and limited dysarthric speech corpora are currently available.

Two possible ways to overcome these problems are 1) using alternative informative features in addition to the acoustic ones to improve the ASR performance; 2) fine-tuning an ASR system based on a large dataset. Note that both approaches are not limited to the specific clinical landscape of dysarthria, but can be promptly transferred to other, less focused, application contexts, and are worth to be implemented *per se*. Another possibility is to take on the problem by reducing its complexity. Indeed, several contexts do not require the recognition of a large vocabulary but only of some commands. This would imply a lower number of labels and a consequent reduce amount of necessary training data.

In the following, we briefly introduce the Automatic Speech Recognition systems whose functioning is the core of a good understanding of this thesis. Successively, we expose how we put into practice the aforementioned approaches, adopted to overcome the limits of ASR of dysarthric speech. In short, these are: the use of articulatory information in the ASR system, the adaptation of the ASR model to the dysarthric dataset/speaker, and the reduction of the recognition vocabulary to a list of commands.

1.1 Automatic Speech Recognition (ASR) systems

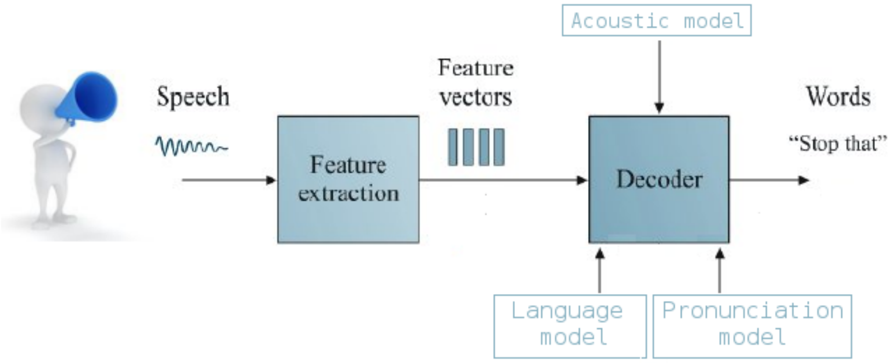


Figure 1.1: The functioning of an ASR system. The speech signal is pre-processed to extract some acoustic features, then used as input for a decoder. The decoding is decomposed into three modules: the acoustic, the language and the pronunciation one. The final output is the text corresponding to the pronounced words.

An automatic speech recognition (ASR) system can be interpreted as a speech-to-text machine, that takes a speech audio as input and returns its transcription. A typical speech recognizer is shown in Figure 1.1. The first step in ASR is the signal processing in which informative feature vectors are extracted and fed into a speech decoder. A common speech representation is given by the Mel-Frequency Cepstrum Coefficients (MFCCs) [15]. A MFCC is defined as the real cepstrum of a windowed short-time signal (called *frame*) derived from the Fast Fourier Transform (FFT) of that signal. The mel-frequency scale is applied to approximate the auditory system. An illustration of the signal processing step is reported in Fig. 1.2.

After extracting the feature vectors, the decoder generates the word sequence $W = \{w_1, \cdots, w_U\}$ that maximizes the posterior probability for the acoustic feature sequence $X = \{x_1, \cdots, x_T\}$, i.e.

$$\max_W P(W|X). \tag{1.1}$$

Problem 1.1 can be re-arranged by following the Bayesian's rule as

$$\max_W \frac{P(X|W)P(W)}{P(X)} = \max_W P(X|W)P(W). \tag{1.2}$$

For large-vocabulary speech recognition systems, directly solving 1.2 can be extremely hard. For this reason, in practice, the optimization problem is decomposed

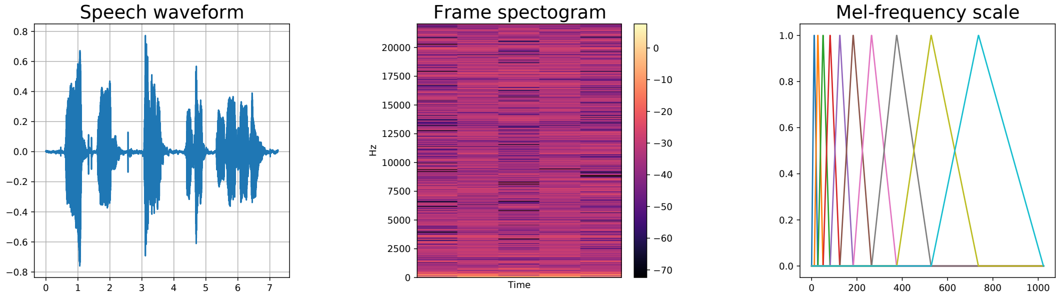


Figure 1.2: An illustration of the feature extraction. (left) A speech waveform example. (center) Spectrogram of a speech window (*frame*). (right) The mel-frequency scale.

in subword models as follow

$$\max_{W} P(X|W)P(W) = \max_{W,S} P(X|S)P(S|W)P(W), \tag{1.3}$$

where $S = \{s_1, \cdots, s_T\}$ is a sequence of word subunits. Phonemes are usually chosen as subunits but other choices may be letters, syllables or demisyllables.

Equation 1.3 decomposes the decoding into three components.The *Language model* $P(W)$ studies what words are likely to co-occur and in what sequence. For example, the sequence of words "I eat pasta" will be more likely than "I eat cars". The *Pronunciation lexicon* $P(S|W)$ provides the sequence of subunits of a word. It may be, for example, a pronunciation dictionary. Lastly, the *Acoustic model* $P(X|S)$ represents the relationship between the audio signal and the linguistic units and, unlike the other decoding components, it directly depends on the set of observed acoustic features X.

In this thesis, we focus on techniques to improve the acoustic model. This is usually based on Deep Neural Networks (DNN) and Hidden Markov Model (HMM) (Fig. 1.3). In such approach, the DNN is trained to classify each single frame and it outputs $p(s_t|x_t)$, $1 \leq t \leq T$. The emission probabilities $p(s_t|x_t)$ are then used by HMM to provide the sequence X maximizing $P(X|S)$. In the supervised framework, the DNN requires frame-level target annotation corresponding to the audio data. This necessitates a further step, called *State alignment*, in which the transcribed data has to be aligned to the speech data by identifying which frames in the feature sequence correspond best to a particular subunit (e.g., phoneme states or senones). A simplified version of this procedure is shown in Fig. 1.4). Typically, alignments are obtained via a Gaussian Mixture Model – Hidden Markov Model (GMM-HMM) system trained with the Baum-Welch algorithm. This is an iterative procedure in which a more accurate alignment is provided at each step.

A clear limitation to all the described phases is that they are optimized separately. In the last decade, the advent of deep learning promoted alternative approaches. End-to-end (E2E) models [16–18] are systems that directly maps the input audio sequence

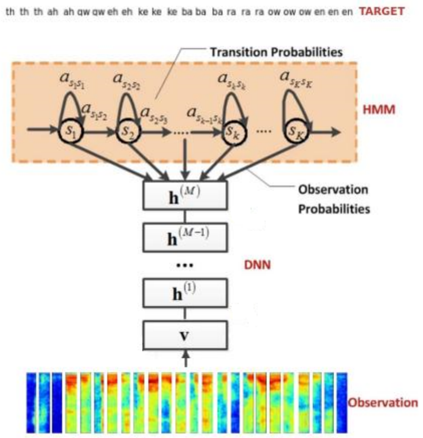

th th th ah ah qw qw eh eh ke ke ke ba ba ba ra ra ra ow ow ow en en en TARGET
Transition Probabilities
HMM
Observation Probabilities
DNN
Observation

Figure 1.3: Typical Acoustic Model: the HMM-DNN architecture. The acoustic vectors x_t are the input of the DNN model that provides the emission probabilities $p(s_t|x_t)$ then used in the HMM model.

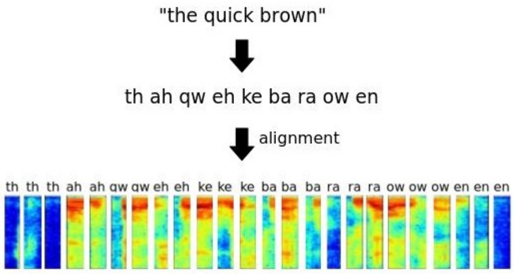

"the quick brown"
th ah qw eh ke ba ra ow en
alignment
th th th ah ah qw qw eh eh ke ke ke ba ba ba ra ra ra ow ow ow en en en

Figure 1.4: A simplification of the state alignment: the text is decomposed in sub-units (phonemes, in this case) that are aligned with the acoustic features. Typically, phonemes are subdivided in states that are then aligned with the acoustic vector.

to the sequence of words or other graphemes. Compared to DNN-HMM, E2E systems offer two main benefits. Firstly, multiple modules are merged into one network for joint training. This results into a reduction of the decoding components and allows to use a loss function that is more relevant to the final global optimization goal. Secondly, as it directly models the text sequence, such a framework does not require a procedure to align data. We are particularly interested in these architectures as the alignment of pathological speech may be not feasible and, a wrong or noisy alignment may have catastrophic consequences on the model training.

Moreover, in Chapter 3 and 4 we will adopt sequence-to-vector training for Voice Command Recognition systems. Such systems can be considered as small-vocabulary ASR and, contrary to the standard ASR, these are based on grammar models rather than language models. Even if the complexity model is reduced, Voice Command Recognition training presents several challenges, including building a grammar resistant to distortions, similar sounds and words inside of commands.

1.2 Speech production knowledge for ASR of dysarthric speech

Intuitively, as the speech impairment mainly involves the speech production issue, taking advantage of articulatory information seems to be a very natural approach. Moreover, adding articulatory representations may compensate for the lack of acoustic data recorded from impaired people. More generally, acoustic-based ASR treats speech as a surface acoustic phenomenon with lexical or phonetic hidden dynamics but without any physical constraints in between. For some complex phenomena observed in speech a strictly acoustic description is not sufficient: however, they can be easily and compactly represented through vocal tract information. For instance, acoustic features model complex acoustic effects of coarticulation, while motor features can provide information at the production level specifying precisely where, when, and how coarticulation occurs.
The integration of speech production knowledge in ASRs is also supported by theories of speech perception (e.g. Motor Theory of speech perception (MTSP) [19]) in which the perception of speech involves the perception of motor gestures and requires access to the motor system. Additionally, neurophysiological studies provided a partial support to MTSP and showed the contribution of the activity of the motor cortex to speech perception [20].

Following the idea of articulatory-acoustic ASR for individuals with speech impairments, preliminary studies have been conducted in [21]. Here, the author adopts the differential entropy H as measure of the degree of statistical disorder in both acoustic and articulatory data for dysarthric and non-dysarthric speakers. It turns out that the acoustics of dysarthric speakers are much more disordered than the ones of non-dysarthric speakers. On the contrary, the difference in articulation entropies between dysarthric and non-dysarthric individuals is unexpectedly small. This means that, despite the motor impairments, dysarthric and non-dysarthric individuals articulate with a similar level of consistency. Thus, compared with the acoustic features, the articulatory ones contain less variability and offer a more robust representation for the ASR of dysarthric speech.

Unfortunately, the recording of the vocal tract (shown in Fig. 1.5) is pretty invasive and costly. Therefore, having access to these measurements is often difficult, in the training phase, and never possible, in the testing phase. It follows that techniques to estimate them are necessary. A standard approach is the Acoustic Inversion (AI), in which an acoustic-to-articulatory mapping is learned. However, only few studies [22–24] focus on the speaker-independent case, and none of these confront the generalization of the mapping across datasets.

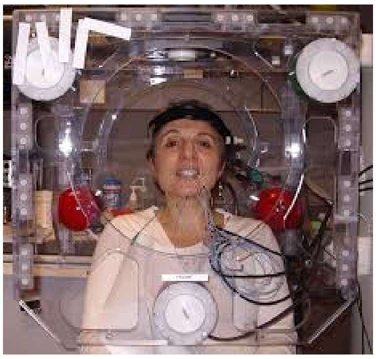


Figure 1.5: Electromagnetic Articulography from TORGO dataset [14].

In this thesis, we aim at overcoming the difficulty in having access to acoustic-articulatory corpora. Specifically, we focus on two approaches to build methods able to generate articulatory features both when motor measurement are available during training and for an audio-only dataset.
First, we look for strategies to improve existing supervised methods (e.g., the AI map) that reconstruct the articulatory features, by integrating articulatory prior knowledge into the model. Secondly, we introduce semi-supervised models that simultaneously leverage articulatory production knowledge, to extract raw articulatory information, and acoustic features, to capture complex phenomena (e.g., the co-articulation effect), in order to synthesize accurate articulatory features.

This study is finally adopted to synthesize articulatory features for an only-speech corpus, and integrate them into the ASR system as secondary target or additional input to the acoustic data. The proposed approach will be detailed in Chapter 2.

This work have been carried out in collaboration with Leonardo Badino and Raffaele Tavarone [25].

1.3 Adaptation strategies for dysarthric speech

One of the major issues in ASR for pathological speech is the impossibility of collecting a large enough dysarthric-speech dataset. One possible alternative is the development of an ASR device based on a large dataset (e.g., a healthy-speech dataset) and, as a second step, the fine-tuning of the system to adapt it to the dysarthric speech. More generally, this procedure is known in machine learning as *domain adaptation* [26, 27] or *transfer learning* [28] and it is required in presence of a mismatch between the distributions of the training data (*source*) and testing data (*target*). As shown in Fig. 1.6, this can entail a poor performance on the target data.

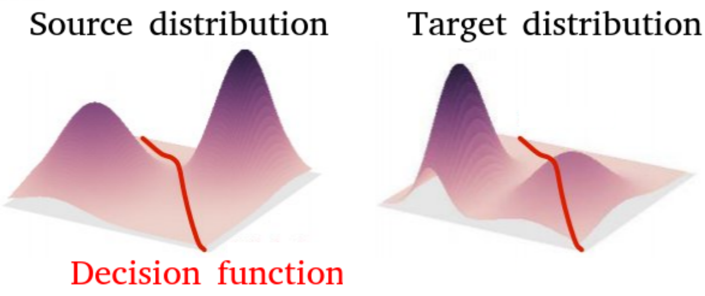


Figure 1.6: An example of mismatch between the source and target distributions that affects the model performance.

In speech recognition, the mismatch may be the result of changing recording conditions or different acoustic environmental noises.
Moreover, a training-testing mismatch can also occur due to the speaker differences (e.g., different vocal tracts, different accents) and speaking styles (e.g., speaking rate). This is even more pronounced for dysarthric speakers, who show different speech features based on the type and the degree of their disorder. The process in which the ASR is tuned to match the characteristics of a speaker is referred to as *speaker adaptation* [29–31].

In Chapter 3, we confront the source-target mismatch problem in a context of dysarthric speech and, specifically, we consider domain adaptation for dysarthria detection and speaker adaptation for ASR. In both cases, we deal with the condition in which multiple sources (i.e., multiple datatsets or speakers) are available. To address these issues, we propose an algorithm based on the Optimal Transport (OT) Theory [32] that simultaneously estimates the similarity between each source and the target distributions and, based on this similarity, learns a classifier for the target domain. The distributions similarity is measured by the Wasserstein distance. The unique feature of this approach is that, as well as performing the adaptation, it provides a closeness measure between source and target that has two main benefits. First, it allows to select only the relevant sources and discard the useless or misleading ones. Second, it helps us to interpret the data as, for example, in the case of speaker adaptation it provides information about the speakers similarity.

To the best of our knowledge, this is the first attempt of applying OT-based techniques on speech tasks. We chose to rely on the Wasserstein distance as it has several advantages over the other commonly used metrics, such as the Kullback-Leibler (KL) divergence. Among the others, the Wasserstein distance does not require both distributions to be on the same probability space. For instance, given a Gaussian distribution, if we apply a translation with an increasing shift (the left graph of Figure 1.7), both Wasserstein distance and the KL divergence increase as well. However, when the supports of the distributions do not overlap anymore, as shown in Figure 1.7 (*Right*), the KL divergence explodes to infinity whereas the Wasserstein distance still assumes real values.

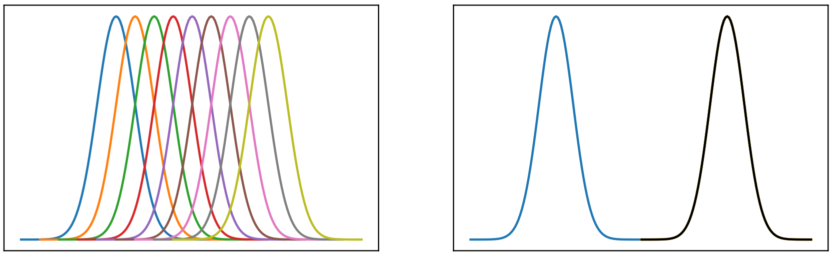


Figure 1.7: 1D-Gaussian distributions obtained by translation. We increased the distributions shift (on the left) until the supports of the distribution do not intersect (on the right).

The MSDA-WJDOT algorithm has been developed in collaboration with Rémi Flamary, Alain Rakotomamonjy and Massimiliano Pontil [33].

1.4 Small vocabulary ASR of dysarthric speech

As mentioned before, there are several scenarios in which only a small vocabulary needs to be recognized. For instance, if we think about controlling a computer by the voice a possible vocabulary may be “turn on”, “turn off”, “open the folder”, etc.

Reducing the number of outputs results in reducing the problem complexity. Moreover, one of the major benefits is that the goal could be achieved by performing commands classification via a sequence-to-one training (i.e., an E2E approach) and, therefore, the phonetic annotation (and, consequently, the speech alignment) is not necessary. This is a not negligible advantage as, while using a reliable phonetic transcription is essential in HMM-DNN approaches, the alignment procedure is not a trivial process and can be particularly hard in presence of dysarthric speech. Recently, several works [34–37] focused on such an approach in which the system is trained to recognize a limited number of commands or key words.

These methods turn out to be particularly suitable in the context of assistive technologies, that are required to be specific and efficient. Even though several Augmentative and Alternative Communication (ACC) devices have recently been developed [38–41], they do not cover all the types of impairment and all the individual needs. Chapter 4 details two projects whose final purpose is a small-vocabulary ASR for specific needs of people affected by dysarthria.

Specifically, we introduce the AllSpeak project whose final product is a mobile application designed for Android. The application is based on an ASR system that recognizes some basic needs, such as “I feel pain”, and acts as communication device for individuals with ALS, especially when they are at the latest stages of the disease. The second project, named EasyCall, addresses the case in which a patient wants to make a call but motor control abnormalities do not allow the normal use of the phone. A suitable solution would be a smartphone Contact application controlled by the voice.

We here move a step towards this goal, by collecting a speech corpus of commands related to the task of making calls and managing a contact mobile application. The dataset includes recordings from both healthy and dysarthric speakers and it represents the largest dysarthric corpus in Italian to date.

Many persons have been involved in the AllSpeak project, from the collection of the speech dataset to the development of the Smartphone Application. These are Cecilia Di Nardi, Alberto Inuggi, Nilo Riva, Ilaria Mauri and Leonardo Badino [42].

Contributed to the EasyCall project Luciano Fadiga, Mariachiara Sensi, Leonardo Badino, Simone Giulietti, Arianna Braccia and Marco Emanuele. A special thank to Elena Zucchini who suggested the name for this project.

Chapter 2

Speech production knowledge for ASR

In this Chapter, we aim at improving the ASR performance by exploiting articulatory information. Hence, we firstly introduce the articulatory features (AFs) and their benefit in ASR, and we briefly reviews methods for recovering them when not available (Sec. 2.1).

Unfortunately, as the data acquisition is invasive and costly, only few and small articulatory corpora are available. Also, it is not feasible an *online* recording of the vocal tract. Consequently, when available, articulatory measurements can be used only during the training phase of the ASR system. In Sec. 2.2, we address this issues by proposing supervised and semi-supervised techniques to generate AFs. We particularly focus on their generalization across dataset as we are interested in synthesizing AFs for speech corpora. All the proposed methods are based on the use of phonetic features, that represent the canonical configuration of the vocal tract during the speech production.

Sec. 2.3 explores two strategies to integrate the synthesized AFs into the ASR system. Finally, Sec. 2.4 discusses the obtained results, and the possible applications of the present study to dysarthric speech. Additional material can be found in the Appendix 2.5.

2.1 Articulatory data and speech production knowledge

In the last decades, researchers deeply studied the speech production mechanism dividing it into four stages: 1) the language processing, in which the content of an utterance is converted into phonemic symbols in the brain's language center; 2) the motor control, in which motor commands are generated in the motor cortex; 3) the articulatory motion, in which the vocal organs (e.g., jaw, lips, or tongue) produce articulatory movements based on the received motor commands; 4) the sound generation, that consists in the emission of air from the lungs in the form of speech.

The movements of the articulators can be directly measured in real time. The most common recording techniques are the Electromagnetic Articulography (EMA) and the x-ray microbeam system. The first induces current in sensor coils, usually placed on different points of the tongue and on other parts of the mouth, to measure their position by alternating magnetic field from fixed transmitter mounted on a helmet. The latter uses 2-3 mm gold pellets attached to the articulators which are tracked by a narrow, high-energy x-ray beam. Both methods provide continuous-valued measurements in the 2D-space, called Pellets Trajectories (PTs). Figure 2.1 shows the EMA recording (on the left) and the data from the x-ray microbeam representing the right profile of the speaker (on the right). In particular, the PTs in figure consist of x-y paths of: upper and lower lips, 4 tongue points, one mandible molar and one mandible incisor.

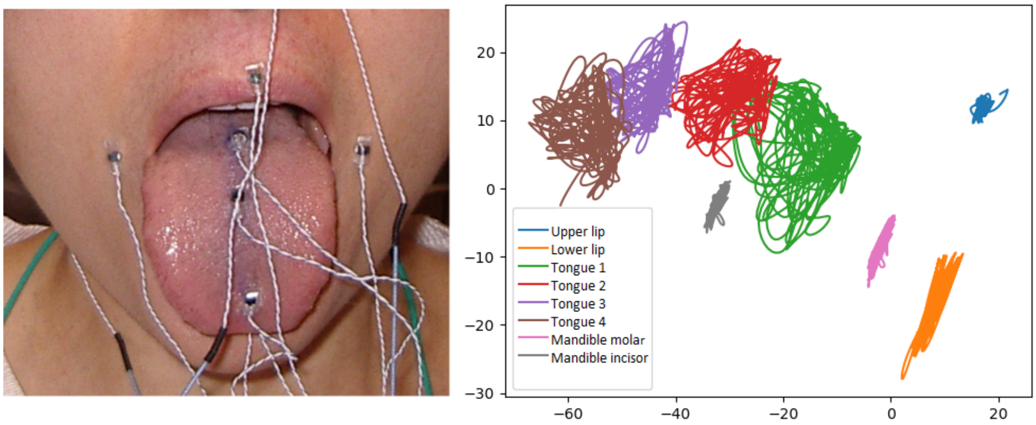


Figure 2.1: EMA recordings during speech production.

Another popular data acquisition technique is the electromyography (EMG) [43] that records the activation potentials of articulatory muscles during the speech production. The drawback of this technique is that EMG signals are usually not directly used but they need additional steps to derive other features depending on the speech task. More recently, other recording methods based on medical imaging have been developed. Among these, ultrasound (US) imaging [44] presents some interesting benefits. It is a low-cost technique and it provides a good temporal sampling (e.g., 50-100 Hz in the case of vocal tract imaging). However, US images can capture information only from the mouth and the higher part of the pharynx, leaving out the rest of the vocal tract. Also, sometimes is not feasible visualizing the tongue due to the jaw bone and the air in the sublingual cavity. Both EMG and US methods have the major advantage that they can also record non-audible speech (i.e. the silence). Thus, these turn out to be particularly useful and employed in speech communication contexts that are noisy or in which audible acoustic signals are not available.

Aside from the actual articulatory data, there are also discrete representations of the speech production covering both categorical features and discretized position (e.g., high/mid/low). One of the most known representations derives from the Articulatory Phonology Theory [45, 46] proposed by Browman and Goldstein in the late 1980s. Contrary to the majority of previous theories, that considered binary features, the Articulatory Phonology aims at describing the vocal tract through articulatory gestural scores. A gestural score is meant to be a certain degree of constriction that occurs in a given location of the vocal tract. The degrees of freedom in articulatory phonology are referred to as vocal tract variables (VTVs) and include the locations and constriction degrees of the lips, tongue tip, and tongue body, and the constriction degrees of the glottis and velum. The advantage of this type of representation is that it does not depend on a specific parametrization of the space.

In this work, we consider eight VTVs: lip protrusion (LP) and aperture (LA), tongue tip constriction location (TTCL) and degree (TTCD), tongue body constriction location (TBCL) and degree (TBCD), velic opening degree (VEL), and glottal opening degree (GLO). The LP represents the (horizontal) position of the lips (they can be protruded or closer to the teeth), while LA is the degree of opening of the lips. Two points on the tongue are considered: the tip and the body. For both of them, the constriction location describes the horizontal position, while the constriction degree measures the distance from the palate. VEL represents the state of the velum (closed or open) and provides information about if the phone is a nasal or non-nasal. Finally, GLO describes the state of the glottis and tells us if the sound is voiceless or voiced.

The VTVs cannot be directly measured but they can be extracted from pellet trajectories by using the transformation procedure described in [47]. LP is computed as the average horizontal position of the upper and lower lips, while LA is given by the difference between the vertical position of the upper and lower lips. The TTCL and TBCL are simply the x-values of the PTs on the tongue. More tricky is instead recovering TTCD and TBCD. The extraction requires the shape estimation of the hard palate, which can be computed by fitting a second-degree polynomial curve to the tongue measurement data as illustrated in Fig. 2.2 (on the left). TTCD and TBCD are defined as the minimum distance of the palate from the tongue tip and tongue body, respectively.

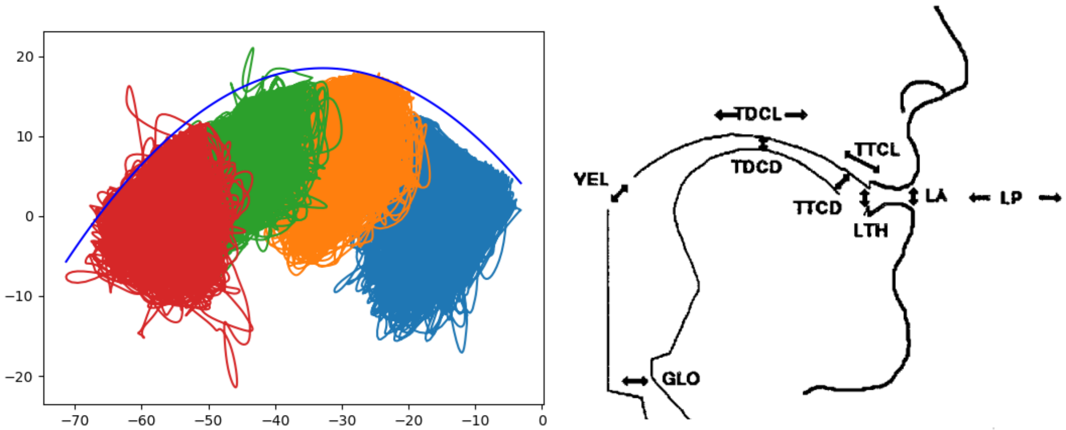


Figure 2.2: Vocal tract variables

Moreover, since articulatory motion data is not easily available, there have been many attempts to extract and model articulatory rules from the data or linguistic studies. We refer to this articulatory information as *speech production knowledge*.

2.1.1 Articulatory ASR

Human speech production requires the integration of diverse information sources: auditory, somatosensory and motor, represented in the temporal, parietal and frontal lobes of the cerebral cortex, respectively. These regions and their interconnections constitute the neural control system responsible for speech production. Taking inspiration from neurophysiology, vocal tract measurements have been provided as complementary information to the acoustic representation in many speech-based devices. They have been proved to be beneficial for automatic speech recognition [48, 49], as well as several other speech technology applications, including speech synthesis [50], pronunciation training [51] and speech-driven computer animation [52]. For the thesis purpose, we only focus on its benefit for the ASR.

Moreover, we want to remark that integrating articulatory information in the ASR may be particularly suitable for the recognition of dysarthric speech. Indeed, several studies [53–57] have showed a direct relationship between the vocal tract features and the spectro-temporal deviations in pathological speech. This resulted in an increasing interest towards dysarthric speech recognizers that incorporate speech production information ([58–61]). Nevertheless, these works are often not applicable as they necessitate of real vocal tract measurements.

2.1.2 Previous works

Typically, articulatory recordings are much more difficult to collect than audio and require extensive pre-processing steps to reduce noise and interpolate missing data [62]. This results in few and relatively small corpora of articulatory data and, as a consequence, in a strong limitation to their use.
Learning a reliable reconstruction of the articulatory features, that generalizes well across speakers and datasets, would allow a more significant use of articulatory information in many applications. In the last decays, most of the studies (e.g., [63–65]) focused on the learning of a mapping between the acoustic and the articulatory space, also known as Acoustic Inversion (AI) map, and on the use the learned articulatory information in an ASR system. While most of these studies have focused on speaker-dependent AI, there is some recent work on the speaker-independent case [22–24]. In [66], the authors propose a statistical method in which they learn a two-steps AI map based on Hidden Markov Models (HMMs).

However, due to the limited amount of data the AI map usually suffers of poor generalization to new speakers and, moreover, the aforementioned studies did not investigate the generalization across datasets, that is instead a crucial issue.

2.2 Estimation of speech production knowledge

In this section we address two questions: (1) can phonetic information, added or substituted to the audio signal, improve the generalization of the AI map across speakers/datasets? (2) Can we generate accurate articulatory features (AFs), starting from some phone-specific prior articulatory knowledge and using very little or zero vocal tract measurements?

To confront these issues, we introduce two phonetic features types and methods based on them, to generate accurate AFs. It is crucial to remark that the final purpose of this work is the integration of synthesized AFs in the ASR system and in such a framework, we have access to the phone labels only during the training phase. Consequently, phonetic features cannot be computed during the testing phase. As we will see in Sec. 2.3, this issue can be easily overcome by learning an additional mapping from the acoustic features to the synthesized AFs, or by using the AFs as secondary target (required only during training).

In the following, we investigate the use of phone labels and two phone-dependent features that we call linguistic and statistical features, respectively. Both of them can be extracted through a look-up table. The linguistic features derive from the Articulatory Phonology theory [45, 46], while the statistical features represent the average configuration of the vocal tract during the emission of a given phoneme. Although the idea of pairing phone labels with input acoustic features to recover AFs is not new [67, 68], here we test the utility of phonetic features in both matched (generalization to new speakers within the same dataset) and mismatched (generalization across datasets) training-testing conditions. The mismatched condition is created by training and validating on male speakers and testing on female speakers, and vice versa. We expect the phonetic information to be particularly helpful in the mismatched case, as it is speaker and environment independent. The two testing conditions are shown in Fig. 2.3.

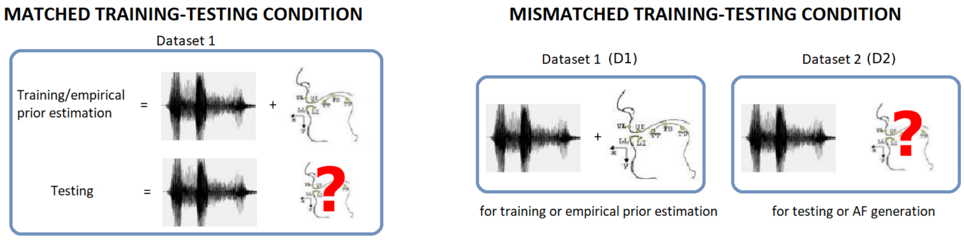

MATCHED TRAINING-TESTING CONDITION
Dataset 1
Training/empirical prior estimation
=
+
Testing
=
?
MISMATCHED TRAINING-TESTING CONDITION
Dataset 1 (D1)
Dataset 2 (D2)
+
?
for training or empirical prior estimation
for testing or AF generation

Figure 2.3: Matched and mismatched training-testing conditions for articulatory reconstruction methods.

To address question 1, we exploit the phonetic features in addition or in substitution to acoustic information in the AI map. Henceforth we will refer to AI and its variants as supervised methods, as articulatory measurements are used as target to train the model and perform the AFs reconstruction. Adding side information, as proposed here, or using adaptation techniques to make AF reconstruction more general may still be very challenging as existing articulatory datasets are small and only cover the read-speech speaking style.

A possible alternative, explored in this thesis, is to estimate AFs directly from audio-only datasets given some prior knowledge about speech production. This alternative strategy addresses our question 2 and the proposed methods are defined as semi-supervised. This approach in principle does not require any articulatory data but some articulatory measurements can still be used to compute or refine the articulatory priors (hence the name “semi-supervised”). In particular, we employ the phone-dependent articulatory priors (the aforementioned linguistic and statistical features) to extract the AFs. We propose three semi-supervised methods, all based on deep neural networks, in which we leverage phonetic and acoustic information to generate latent motor representation of the acoustic data.

This section is organized as follows. In Sec. 2.2.1, we introduce the aforementioned linguistic and statistical features and the procedure to compute them. We then propose their use in deep learning-based methods, described in Sec. 2.2.2 and 2.2.3, to synthesize AFs. We tested the proposed approaches on several experiments detailed in Sec. 2.2.4. Results are reported in Sec. 2.2.5, while a final discussion can be found in 2.2.6.

2.2.1 Phonetic features

We consider the VTVs introduced in 2.1 and two additional binary features, *consonant* and *silence* (1 if the sound is consonant/silence, 0 otherwise). This results in a 10 integer-valued vector, representing the articulatory information during the emission of a phoneme. For the sake of simplicity, in the following we will use the term VTVs to refer to the extended 10-dimensional representation.

In this section, we propose two types of phonetic features containing articulatory priors that can be extracted from two look-up tables by requiring only the phonetic transcriptions. Both tables report estimations of the VTVs during the production of each phoneme. One table derives from linguistic considerations, while the other is the result of a statistical study of articulatory measurements. In both cases, the extracted phonetic features are a raw representation of the vocal tract that, however, does not take into account more complex phenomena, such as the co-articulation effects. In the following we describe the look-up tables and their extracted features.

Linguistic Features (LFs) The author in [69] estimates the VTVs corresponding to the emission of a given phone, based only on observations on some linguistic behavior (e.g., the synchrony constraints on pairs of VTVs). More details can be found in Appendix 2.5, in which we report the complete table (see Tab. 2.13) with the linguistic-based articulatory priors for all phonemes. Finally, for a given phoneme sequence, we can recover the corresponding sequence of articulatory priors. We refer to the extracted phone-dependent features based on linguistic considerations as Linguistic Features (LFs).

Statistical Features (SFs) We propose an alternative representation based on a simple statistical procedure, requiring hence the use of real articulatory measurements. Specifically, given the PTs of a training dataset, the statistical study follows these steps:

1. recovering the VTVs from the PTs;
2. per-speaker Z-normalization of the VTVs;
3. for a given phoneme, computing the mean value of each VTV;
4. rounding the average values to their closest integer.

A schematization of this procedure can be found in Fig. 2.4. Also in this case, the final product is a look-up table (see Table 2.14 in the Appendix Sec. 2.5) with an average number of 4 quantization levels per feature. These priors represent the average configuration of the vocal tract of a speaker during the emission of each phoneme. Again, given the phone labels we can retrieve the corresponding features sequence, that we call Statistical Features (SFs).

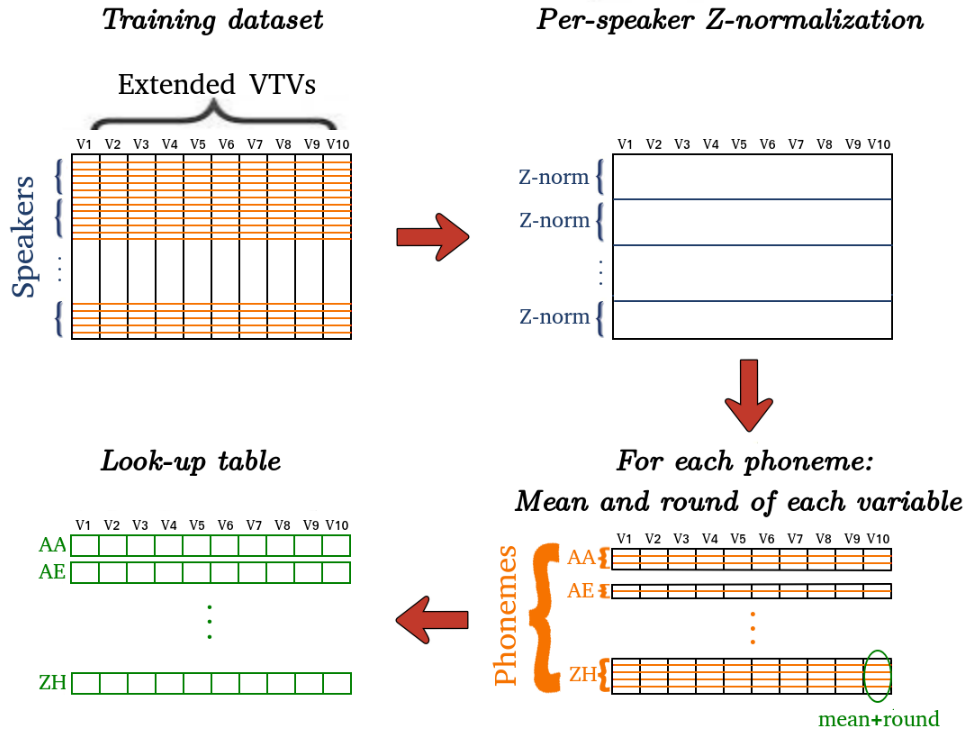
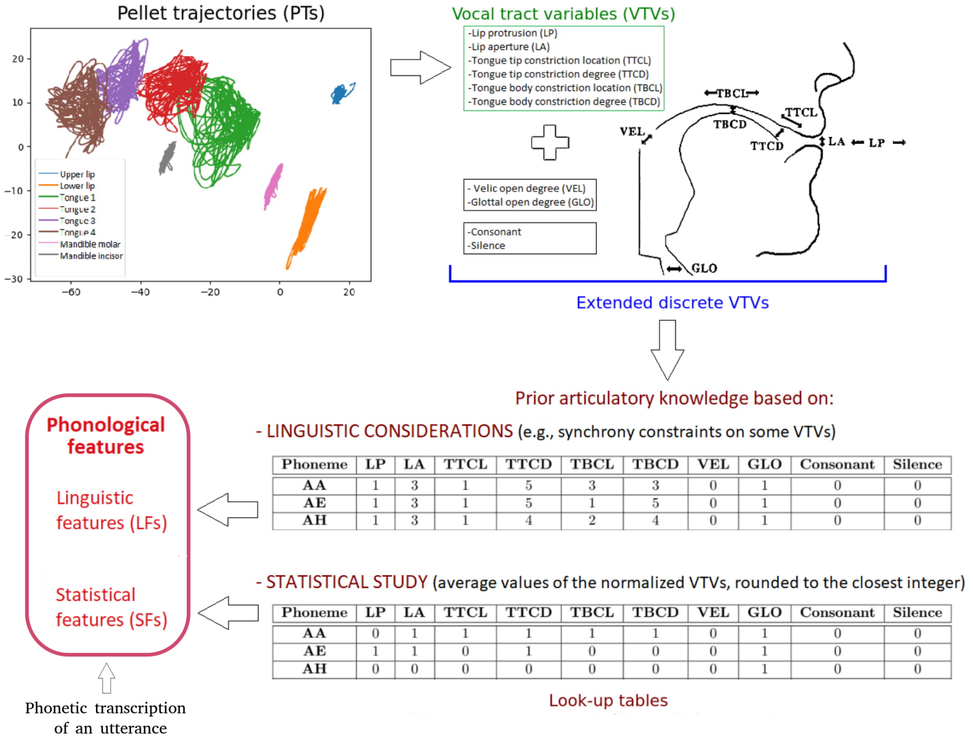


Figure 2.4: Statistical procedure to estimate the motor gestures. The training dataset consists of 10-dimensional VTVs and it can be seen as set of many subsets corresponding to different speakers. Firstly, each subset is Z-normalized. Then, the dataset is reorganized in order to gather together representations of a same phoneme. Finally, each VTV is considered separately. The mean value of every feature during the emission of a given phoneme is then computed and rounded the the closest integer.

Both the linguistic and the statistical features lie within a fixed discrete range. The advantage of these feature types is that they contain phonetic and articulatory information and they can be easily extracted from a look-up table. Indeed, the procedure to compute linguistic and statistical features (summarized in Fig. 2.5) from articulatory data does not need to be reproduced again. For any new dataset, only the phonetic transcription is required to recover the linguistic or statistical features.

As mentioned above, these features refer to a "stereotypical"/phonological description of a phone where the coarticulation effects are not taken into account. This motivated us to develop methods that exploit the linguistic/statistical features for generating a more accurate representation in which complex phenomena, including the coarticulation, are considered.


Pellet trajectories (PTs)
20
10
0
−10
−20
−30
−60
−40
−20
0
20
Upper lip
Lower lip
Tongue 1
Tongue 2
Tongue 3
Tongue 4
Mandible molar
Mandible incisor
Vocal tract variables (VTVs)
-Lip protrusion (LP)
-Lip aperture (LA)
-Tongue tip constriction location (TTCL)
-Tongue tip constriction degree (TTCD)
-Tongue body constriction location (TBCL)
-Tongue body constriction degree (TBCD)
TBCL
TTCL
TBCD
TTCD
VEL
LA
LP
GLO
- Velic open degree (VEL)
-Glottal open degree (GLO)
-Consonant
-Silence
Extended discrete VTVs
Prior articulatory knowledge based on:
Phonological features
Linguistic features (LFs)
Statistical features (SFs)
Phonetic transcription of an utterance
Look-up tables

- LINGUISTIC CONSIDERATIONS (e.g., synchrony constraints on some VTVs)

Phoneme	LP	LA	TTCL	TTCD	TBCL	TBCD	VEL	GLO	Consonant	Silence
AA	1	3	1	5	3	3	0	1	0	0
AE	1	3	1	5	1	5	0	1	0	0
AH	1	3	1	4	2	4	0	1	0	0

- STATISTICAL STUDY (average values of the normalized VTVs, rounded to the closest integer)

Phoneme	LP	LA	TTCL	TTCD	TBCL	TBCD	VEL	GLO	Consonant	Silence
AA	0	1	1	1	1	1	0	1	0	0
AE	1	1	0	1	0	0	0	1	0	0
AH	0	0	0	0	0	0	0	1	0	0

Figure 2.5: The procedure to extract phonological features.

2.2.2 AF estimation: supervised methods

In the supervised methods, we have access to the articulatory data, the audio recordings and the phonetic annotations. Here, we focus on methods to learn a mapping from acoustic features and/or phonetic features (i.e., phone labels or LFs or SFs) to AFs (either in the form of PTs or VTVs).

Firstly, we implement the classical Acoustic Inversion (AI) map in which the acoustic and the articulatory representations are input and target, respectively, to a regressor (e.g., a DNN). For this mapping, phone labels are not necessary either in the training or in testing phase. We then compare this standard approach with the following ones:

- AI map with side information: we exploit the use of phonetic features (phone labels, LFs, SFs) as additional input to the audio signals;
- AFs reconstruction from side information only: we do not consider the acoustic features and we only use the phonetic ones as input for the model. We expect this method to perform worse than the AI map with side information, mentioned above, in a matched training-testing scenario but we could see a better generalization across datasets even if it uses less information. Indeed, acoustic features are largely dependent on the recording conditions (e.g., microphone, recording room, noise, etc.) and, therefore, supervised methods based on the audio recordings may be not robust to the variability across datasets.

2.2.3 AF estimation: semi-supervised methods

The scarce availability of articulatory data motivates approaches that attempt to generating AFs without actually measuring them.

In this strategy the available articulatory information consists of some prior concise description of the typical vocal tract configuration of each phone. Specifically, we propose to use the Linguistic Features and the Statistical Features described in Sec. 2.2.1. Further, we experiment with SFs extracted from both multiple-speaker data and single speaker-data.

We aim at generating AFs from the linguist/statistical features and the acoustic information to simultaneously capture articulatory prior knowledge, from the first, and phonetic-context dependencies, from the latter. Specifically, we propose three semi-supervised methods based on deep auto-encoders [70, 71] or residual networks [72].

Notation We denote by $\mathbf{x}$ the acoustic feature vector, by $\hat{\mathbf{x}}$ the reconstructed acoustic feature vector, by $\mathbf{z}$ the articulatory prior vector (i.e., SFs or LFs) and by $\hat{\mathbf{z}}$ the generated AF vectors. The precision of the generated articulatory features is evaluated by comparing $\hat{\mathbf{z}}$ with measured articulatory features.

Autoencoder-based methods An autoencoder (AE) is an artificial neural network architecture that aims at reconstructing its input through a latent representation (encoding). It consists of two parts: a mapping from the input to the latent representation (encoder, e), and the input reconstruction starting from the encoding (decoder, d). As speech production and speech perception are strictly connected, the autoencoder scheme turns out to be particularly suitable to model the link between acoustic and articulatory features.

Typically, the model is learned by minimizing the input reconstruction function. However, the standard approach ensures us to have one latent representation of the input but there is no guarantee to recover the representation of interest. To force the network to learn a specific encoding, we learn the input reconstruction function simultaneously with the encoding reconstruction loss. Therefore, we add a secondary loss to the input reconstruction one, in which we minimize the ℓ_2 distance between the encoding and the desired representation. In Fig. 2.6 we show the network architecture. In green, we point at the input (on the left) and its reconstruction (on the right). In red, we have the learned encoding (at the top) and the latent representation we force to resemble to (at the bottom).

In the following, we explore two different variants of the AE. In one case, the input reconstruction loss aims at reconstructing the audio signal, while the encoding reconstruction loss involves the phonetic features. In the the second scenario, we invert the role of the acoustic and phonetic features: the AE takes the LFs/SFs as input and reconstructs them by passing through a latent representation that is forced to be close to the acoustic one.

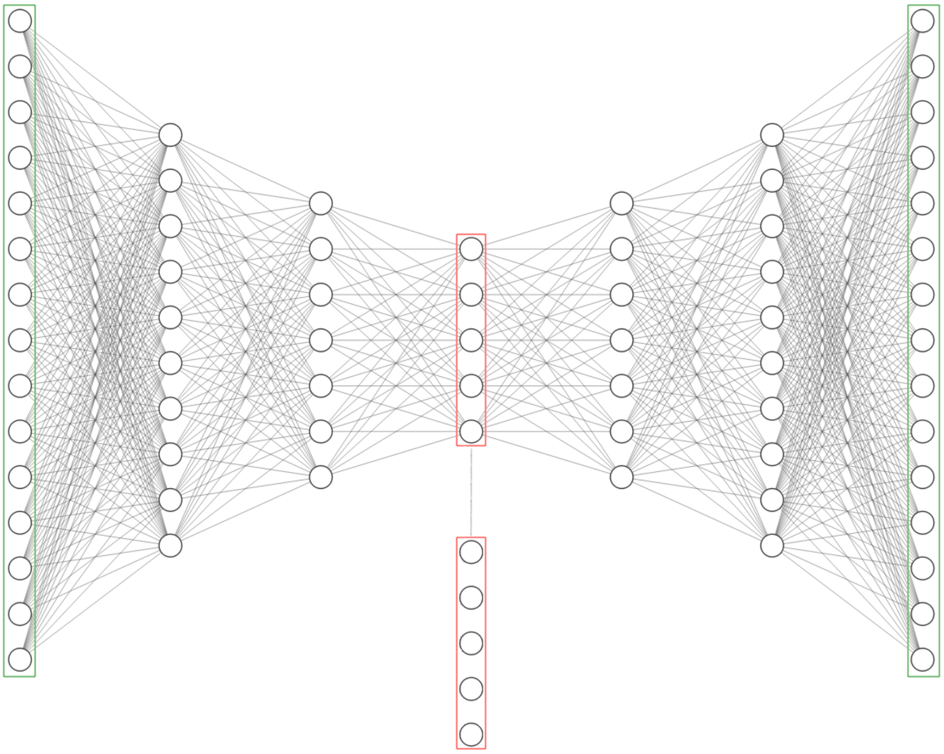


Figure 2.6: Autoencoder architecture with double loss on the input and embedding reconstruction.

- **Autoencoder 1 (AE1)** The first model simulates the neurophysiological behavior: since the vocal tract movements are the physical causes of spoken sounds, we can interpret the motor vector as a latent representation of the acoustic signal. Hence, AE1 takes the audio as input and returns its reconstruction. This map goes through the encoding layer, which we encourage to imitate an articulatory representation by adding an additional loss function.

Let $\mathbf{z}_t \in \mathbf{R}^{10}$ be the vector of linguistic or statistical features at time t, and $\mathbf{x}_{t-T}^{t+T} = [\mathbf{x}_{t-T}, \ldots, \mathbf{x}_t, \ldots, \mathbf{x}_{t+T}]$ the input concatenation of the audio vectors, where T is the context window length on each side. The objective function at time t is:

$$L_{A1,t} = \underbrace{\| \mathbf{x}_{t-T}^{t+T} - \hat{\mathbf{x}}_{t-T}^{t+T} \|_2^2}_{\text{Input reconstruction loss}} + \lambda_z \cdot \underbrace{\| \mathbf{z}_t - \hat{\mathbf{z}}_t \|_2^2}_{\text{Encoding loss}}, \tag{2.1}$$

where $\hat{\mathbf{z}}_t = e(\mathbf{x}_{t-T}^{t+T})$, $\hat{\mathbf{x}}_t = d \circ e(\mathbf{x}_{t-T}^{t+T})$ and λ_z is a scalar hyperparameter that weights the importance of the second term of the loss. In other words, we force the latent representation of the acoustic features $\mathbf{x}$ to be close to the typical configuration taken by the vocal tract when the phoneme associated to $\mathbf{x}$ is produced. $\mathbf{z}$ can be seen as the mean of a prior multivariate Gaussian distribution, while we do not make any prior assumption regarding its covariance (contrary to variational autoencoders [73]). Here we are assuming that the actual AFs are normally distributed around $\mathbf{z}$. This is supported by qualitative analysis we have carried out per each phone.

- **Autoencoder 2 (AE2)** In the second variant, we reverse the AE structure previously described. Now, $\mathbf{z}$ is the input of the AE which provides the articulatory reconstruction $\hat{\mathbf{z}}$. We force the encoding layer to match the acoustic latent representation $\mathbf{x}$. In this context, the idea is that the linguistic/statistical features are a raw representation of the articulatory features that does not take into account more complex phenomena, such as the co-articulation. Hence, we aim at modulating the phonetic features by taking advantage of the acoustic features that contain more complex information.

In AE2, the loss function to be minimized at time t is:

$$L_{A2,t} = \underbrace{\| \mathbf{z}_{t-T}^{t+T} - \hat{\mathbf{z}}_{t-T}^{t+T} \|_2^2}_{\text{Input reconstruction loss}} + \lambda_x \cdot \underbrace{\| \mathbf{x}_t - \hat{\mathbf{x}}_t \|_2^2}_{\text{Encoding loss}}, \tag{2.2}$$

where $\hat{\mathbf{x}}_t = e(\mathbf{z}_{t-T}^{t+T})$, $\hat{\mathbf{z}}_t = d \circ e(\mathbf{z}_{t-T}^{t+T})$ and $\lambda_x \in \mathbb{R}$ is an hyperparameter. Note that here the articulatory reconstruction $\hat{\mathbf{z}}$ is not a direct function of the acoustic features, as in AE1, but of the phonetic features.

Residual-based method In this approach a deep neural network with one residual layer (ResDNN) takes articulatory prior vectors $\mathbf{z}$ as input features and targets acoustic features (Figure 2.7). The residual layer [72] modulates the input $\mathbf{z}$ with its left and right context weighted by a learned parameter, thus returning a coarticulation-modulated version of the $\mathbf{z}$.

Formally, the output of each i-th element of the residual layer $\hat{\mathbf{z}}_t$ is defined as:

$$\hat{z}_t^i = z_t^i + R_t, \quad R_t = f\left(\sum_{s=t-T}^{t+T} \sum_{j=1}^{10} z_s^j w_{sj}^R \right). \tag{2.3}$$

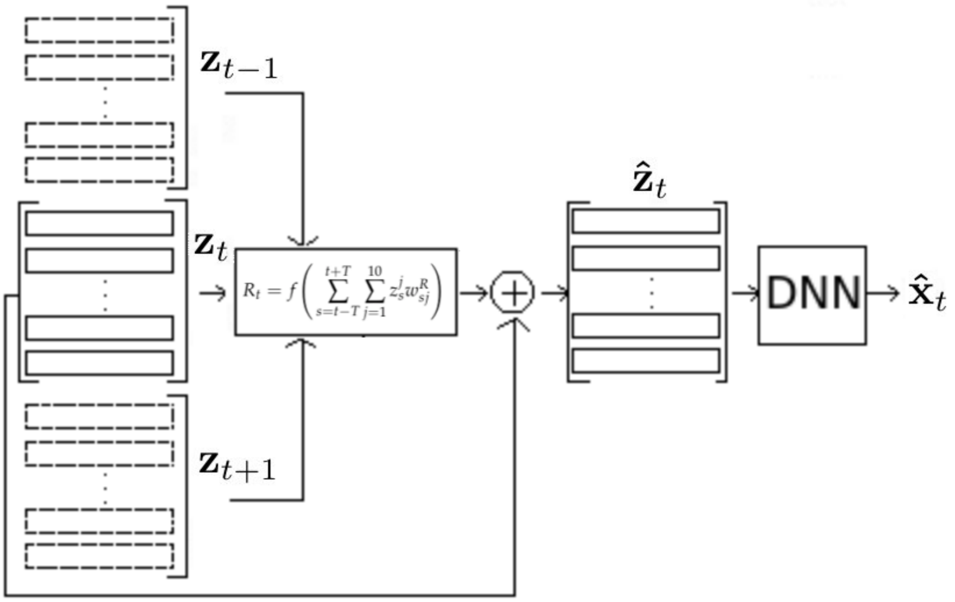


Figure 2.7: Residual DNN structure. The frame context is only used in the residual layer. In this simplified example, T is equal to 1.

R_t is the residual at time t, and the weights w_{sj}^R are the trainable parameters of the residual network. The sums taken over time and features model co-articulation effects. The network is trained to minimize the following loss function:

$$L_{R,t} = \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2^2 + \lambda_w \|\mathbf{w}^R\|_2^2, \tag{2.4}$$

where $\hat{\mathbf{x}}$ is the reconstructed audio, λ_w controls the penalization term, and $\mathbf{w}^R \in \mathbb{R}^{10\cdot(2T+1)}$.

2.2.4 Experimental setup

Dataset All the experiments are carried out on the 47 American English speakers subset of XRMB used in [23, 24], with the only exception that we discard speaker *JW33* (used for validation in [23, 24]), as we discovered some corrupted audio (while we keep speaker *JW58* which was removed in [24] and only remove some corrupted utterances). Articulatory data consists of x-y trajectories of: upper and lower lips, 4 tongue points, one mandible molar and one mandible incisor. Articulatory features is pre-processed as in [23], while acoustic features are the first 13 MFCCs, computed every 10ms from 25ms Hamming windows, plus deltas and delta-deltas. Both acoustic and articulatory features are per-speaker z-normalized.

For the training-testing matched condition we split the dataset into disjoint sets of 35/7/4 speakers for training/validation/testing respectively.

For the training-testing mismatched condition we split the dataset by gender. We refer to the so-obtained subsets as *Male* and *Female* , with 22 and 24 speakers respectively. For supervised methods, when *Female* is used as testing dataset, *Male* is split into 18/4 speakers for training/validation respectively. In the opposite case, *Female* is split into training and validation, with 19 and 5 speakers respectively.

Neural Networks

- Supervised methods are based on bidirectional LSTMs (BLSTMs). The networks have 5 layers each containing 250 memory blocks, with peephole connections and hyperbolic tangent activation function. All the experiments are carried out using Adaptive Momentum Optimizer [74], a piecewise constant learning rate with initial value set to 0.1, a 0.9 momentum, a small constant for numerical stability $\epsilon = e^{-8}$ and initial decay rates of first and second moments 0.9 and 0.999, respectively. Weights are initialized with Xavier initialization [75]. Early stopping is applied to determine the number of training epochs.

- In all the semi-supervised methods, the network input consists of the central vector plus $T = 12$ context vectors per side. Training is performed with stochastic gradient descent. Learning rate exponentially decays every 10000 steps, with initial value 0.01 and 0.96 decay rate. Training is performed for 50 epochs or stopped earlier if the acoustic feature reconstruction error does not decrease.

  Both AE types have a hourglass shape, symmetric w.r.t. the encoding layer. Each encoder (as well as the decoder) has 3 layers with 200, 130, 70 nodes respectively, decreasing towards the encoding layer which has 10 nodes in AE1 and 39 nodes in AE2. Again we use Xavier initialization.

  The ResDNN has 4 layers with 1000 nodes each, while the residual layer has 10 nodes. We fix $\lambda_w = 0.01$ and grid-search the remaining hyper-parameters, based on the audio reconstruction.

We evaluate all methods by computing the average (over features) root mean squared error (RMSE) and the average Pearson's correlation coefficient (r) between per-speaker z-normalized reconstructed and measured AFs (so RMSE is a normalized RMSE).

2.2.5 Results

Preliminary study on the residual function of ResDNN As we do not have any hypothesis on the residual function f of Eq. 2.3, we tested several different functions: the zero function, the identity function, the sigmoid function (with different scaling parameters), and the hyperbolic tangent. Note that for $f \equiv 0$ the first hidden layer coincides with the input layer. In Tab. 2.1, we report the acoustic and motor reconstruction performance provided by the tested functions, for $\lambda_w = 1$. As we can see, the highest correlation is achieved by the identity function. The same conclusion can be drawn from Fig. 2.8 and 2.9 that report the measured and the reconstructed LP and

LA features, respectively. Hence, in the following we will adopt f =identity function.

$f(x)$	r of Audio	r of VTVs
$f(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$	0.487	0.508
$f(x) = \frac{2}{1+e^{-x}}$	0.487	0.513
$f(x) = \frac{2}{1+e^{-(x\cdot 0.25)}}$	0.486	0.497
$f(x) = \frac{2}{1+e^{-(x\cdot 0.075)}}$	0.486	0.491
$f(x) = x$	0.488	0.526
$f(x) = 0$	-	0.483

Table 2.1: Audio and Articulatory features Reconstruction measured by the Person's Correlation at varying of the residual function f.

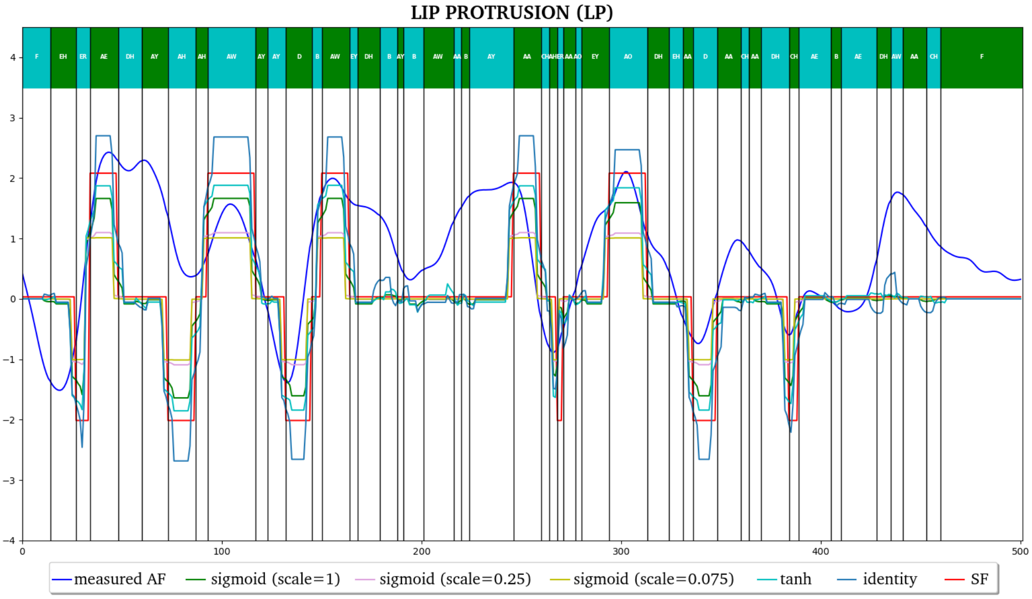


Figure 2.8: Lip Protrusion reconstruction performed by the ResDNN with different residual functions f. We compare the reconstructed feature with the real measurement (in blue) and the statistical feature (in red), that coincides with the case $f \equiv 0$.

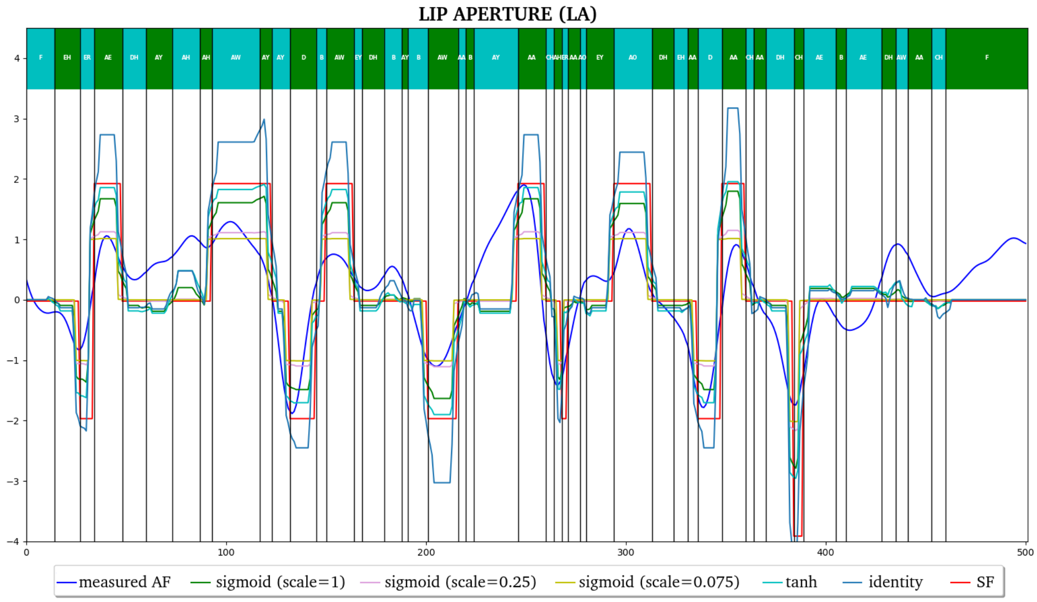


Figure 2.9: Lip Aperture reconstruction performed by the ResDNN with different residual functions f. We compare the reconstructed feature with the real measurement (in blue) and the statistical feature (in red), that coincides with the case $f \equiv 0$.

Matched conditions

- **Supervised methods** In Table 2.2, we compare the average RMSE and correlation for PT and VTV reconstruction of different BLTSM inputs. BLTSM training and evaluation were repeated twice, with different random initialization. To keep tables more readable we only report the mean, the standard deviation is always less than 0.01.

	PTs		**VTVs**	
Input	RMSE	r	RMSE	r
MFCCs (S1)	0.894	0.448	0.879	0.517
MFCCs	0.685	0.721	0.646	0.777
Phonemes	0.664	0.742	0.617	0.782
LFs	0.672	0.732	0.611	0.797
SFs	0.667	0.744	0.618	0.783
MFCCs + Phonemes	0.654	0.757	0.606	0.797
MFCCs + LFs	0.657	0.748	0.602	0.801
MFCCs + SFs	0.655	0.752	0.606	0.798

Table 2.2: Supervised methods results on the test set for PT and VTV reconstruction in the matched condition case. MFCCs (S1) refers to a BLSTM trained on 1 single speaker data (JW14).

Results suggest that all the phonetic features (phone labels, LFs and SFs) outperform MFCCs, and, surprisingly, MFCCs slightly improve reconstruction when combined with phonetic features, despite MFCCs containing much more detailed information than the phone-dependent features. LFs and SFs do not produce relevant improvement w.r.t. phone labels. Table 2.2 also shows AI results when only one speaker is used for training in order to quantify the gap w.r.t. multi-speaker training data and to compare with semi-supervised methods in a limited articulatory data setting.

- **Semi-supervised methods** Table 2.3 summarizes the results provided by the three proposed approaches in the matching conditions setting. For comparison, we also report the performance of the *Baseline* model, where the linguistic and statistical features are directly compared with measured AFs. Again, all the experiments are carried out twice (standard deviation < 0.01). Although LFs and SFs have a similar number of quantization levels, SFs largely outperform LFs in all methods. Most importantly, the generated AFs $\hat{\mathbf{z}}$ always correlate more with actual AFs than the priors $\mathbf{z}$, with the exception of method AE1. That means that AE2 and ResDNN successfully transform the original prior articulatory information into articulatory features that are closer to the actual AFs. AE2 is the most effective method.

	Baseline		**ResDNN**		**AE1**		**AE2**	
Features	RMSE	*r*	RMSE	*r*	RMSE	*r*	RMSE	*r*
LFs	-	0.366	-	0.360	-	0.330	-	0.390
SFs	0.858	0.524	1.010	0.554	0.862	0.507	0.820	0.571
SF1s	0.888	0.514	1.117	0.537	0.876	0.508	0.835	0.563
SF2s	0.872	0.519	1.102	0.524	0.894	0.492	0.826	0.568

Table 2.3: Semi-supervised methods results on the test set. SF1s and SF2s refer to the statistical features computed on the JW14 and JW14+JW12 articulatory data, respectively.

To show that SFs generalize well across speakers, we re-computed the SFs based on only one or two training speakers (SF1s and SF2s) and repeated the semi-supervised experiments. Interestingly, the results obtained with SF1s and SF2s do not considerably differ from SFs. This implies that the statistical representations calculated on few speakers (or just one!) are sufficient to characterize the vocal tract of any other speaker. Importantly, in this limited data setting, ResDNN and AE2 outperform the best supervised method (e.g., $r = 0.537$ and $r = 0.563$ vs. $r = 0.517$). Note that Table 2.3 shows the best AE1 and AE2 performances on the validation set, achieved by fixing λ_z and λ_x at 2 and 0.5, respectively. We did not report the RMSE for the LFs, as they do not reflect the real measurements of the articulatory data. More detailed results can be found in Table 2.4, where the best AE2 performance is reported for two test speakers and for each VTV.

		LP	**LA**	**TTCL**	**TTCD**	**TBCL**	**TBCD**
JW48	RMSE	0.825	0.859	0.838	0.753	0.816	0.828
	r	0.600	0.519	0.590	0.680	0.581	0.563
JW53	RMSE	0.781	0.842	0.845	0.666	0.739	0.745
	r	0.688	0.548	0.581	0.747	0.681	0.686

Table 2.4: Details of AE2 performance for speakers JW48 and JW53 (matched conditions).

Mismatched conditions

- **Supervised methods** Table 2.5 shows the results of the supervised models in the training-testing mismatched conditions. The most striking result is that MFCCs not only perform significantly worse than SFs but also deteriorate the performance of the SFS when combined with them. This is due to the strong speaker dependency of MFCCs (despite their per-speaker normalization), that may be alleviated through speaker adaptation. In Fig. 2.10, we plot 6 graphs reporting the sequence reconstruction of each VTV corresponding to a spoken sentence, when only the SFs are used in the supervised model. We also exhibit the sequence of the SFs (i.e., the model input) and the measured AFs (i.e., the model target).

Input	**Test gender**	RMSE	*r*
MFCCs	*Male*	0.848	0.592
SFs	*Male*	0.604	0.782
MFCCs + SFs	*Male*	0.685	0.743
MFCCs	*Female*	0.860	0.557
SFs	*Female*	0.625	0.787
MFCCs + SFs	*Female*	0.686	0.748

Table 2.5: BLSTM cross-gender VTV reconstruction.

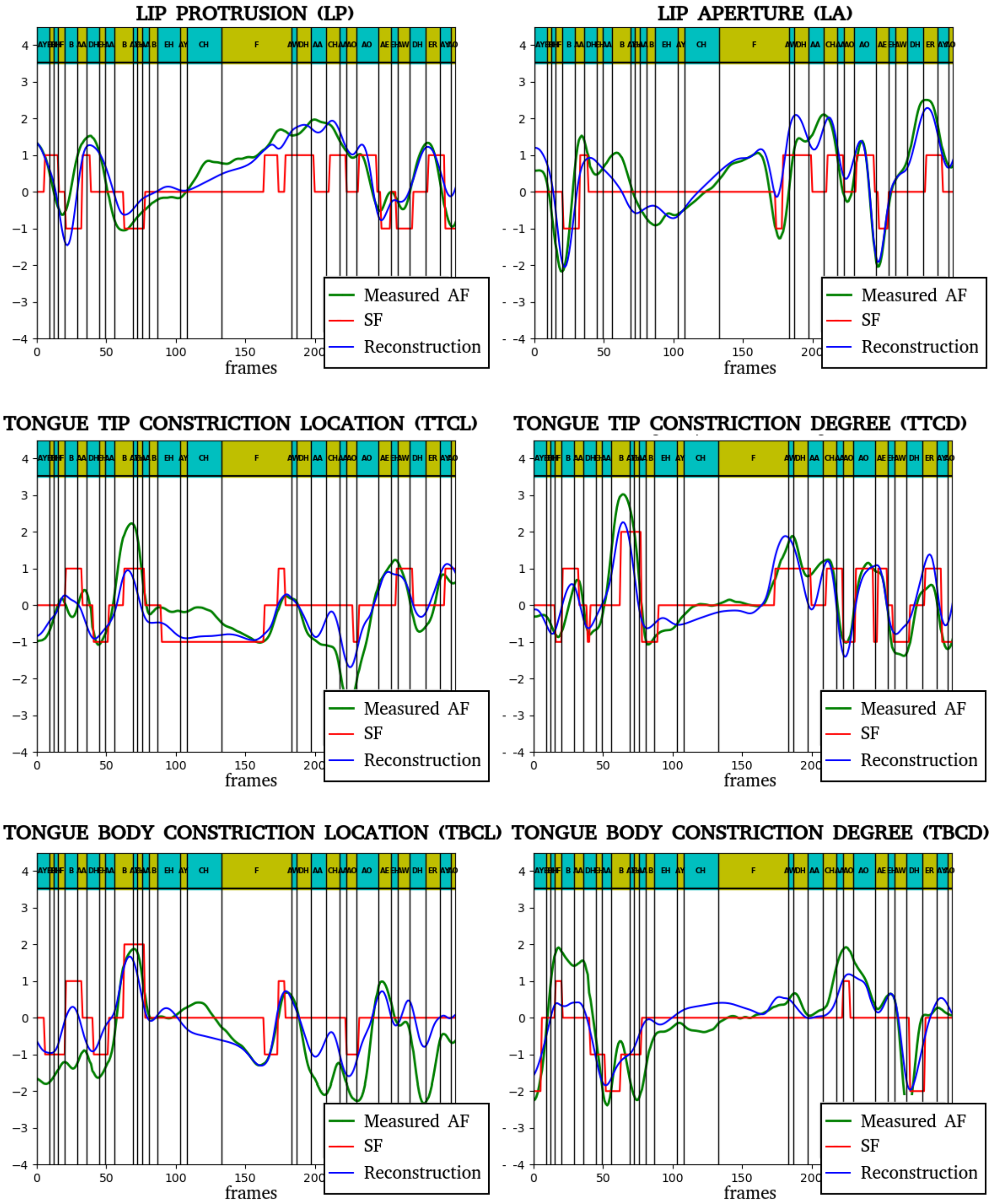


Figure 2.10: VTVs reconstructed by the SFs-to-VTVs map. The reconstruction (in blue) is compared with the SF (in red) and the measured motor gesture (in green).

- **Semi-supervised methods** We only test the AE2 model, as it was the most effective one in the training-testing matching condition. Note that, in this case, AE2 is trained and tested on the same speakers (e.g., *Female*), while priors are computed on a different dataset (e.g., *Male*). Results in Table 2.6 show that (i) AE2 almost matches the supervised method with MFCC; (ii) even in the mismatched case, AE2 reconstruction is not affected by a reduction of articulatory data to a single speaker. In Fig. 2.11, we report the estimated articulatory representation of a sentence spoken by a female speaker (while the AE2 is trained on *Male* data). As we can see, the SFs are smoothed and modulated in a way that provides a reconstruction that better matches the motor measurements.

	Baseline		**AE2**	
Test gender	RMSE	*r*	RMSE	*r*
Male	0.854	0.539	0.816	0.586
Male (S1)	0.877	0.526	0.822	0.579
Female	0.858	0.529	0.821	0.576
Female (S1)	0.867	0.529	0.819	0.576

Table 2.6: Cross-gender evaluation of AE2. *Male (S1)* and *Female (S1)* refer SFs computed from female speaker JW14 and male speaker JW12, respectively.

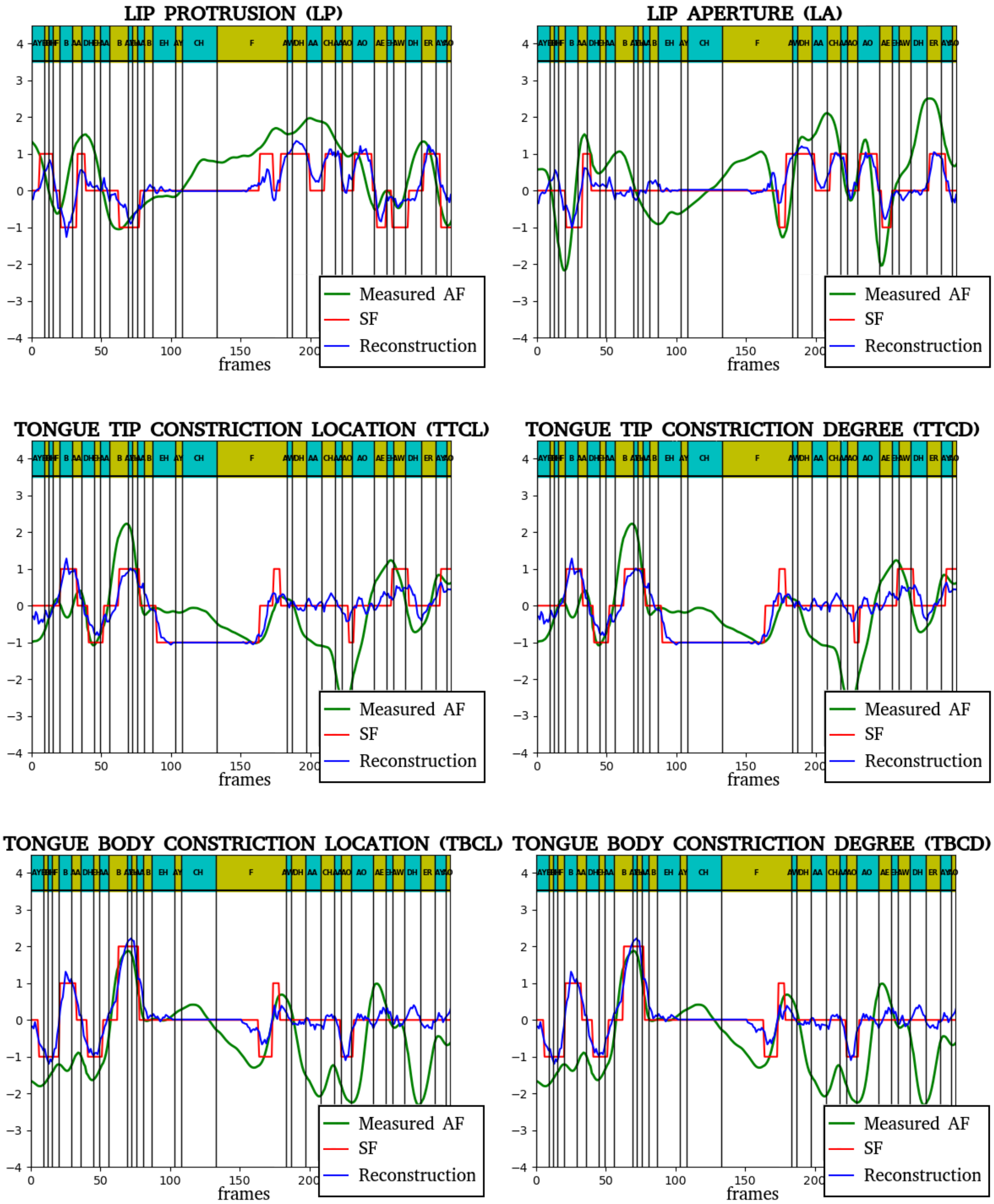


Figure 2.11: AF reconstruction performed by AE2 in the unsupervised way. The reconstructed AF (in blue) is compared with the measured one (in green) and the SF (in red).

2.2.6 Discussion

In Sec. 2.2, we developed methods to synthesize AFs both when we have access to motor measurements (i.e., in a supervised framework) and when only few or zero actual AFs are available (i.e., in a semi-supervised framework). All the approaches are based on phonetic features. Specifically, we investigated the use of Linguistic Features (LFs), introduced in [69], in which articulatory information is extrapolated by linguistic rules and observations. In addition, we proposed a new type of features, called Statistical Features (SFs), that represent the average configuration of the vocal tract during speech production. These are computed via a statistical procedure, resulting in a look-up table providing, for all phonemes, the corresponding average values of the AFs.

A standard supervised approach is learning an acoustic-to-articulatory mapping, known as Acoustic Inversion (AI) map. This work exploited the use of the aforementioned phonetic features, as well as phone labels, as side information in such a mapping. Further, we proposed to substitute the acoustic representation with the phonetic one in the AI map. Although this mapping uses less information than the AI with side information, we found that our proposed methods have comparable performances in the generalization ability across speakers. Also, compared with the AI map, adding phonetic features always improves the reconstruction performance. We further evaluated the supervised methods in a mismatched training-testing condition, i.e. we evaluated their generalization across datasets. The most interesting result is that the best reconstruction is provided by the models in which only phonetic features are used as input. This highlights the robustness of this type of features to the variability across datasets, whereas acoustic features are known to be dependent on the recording conditions.

We then studied semi-supervised methods based on phonetic and acoustic features to generate AFs, when articulatory measurements are not available. We refer to them as semi-supervised rather than unsupervised as the statistical features involve the statistic of real articulatory data. The intuition behind the proposed semi-supervised approaches is that the phonetic features contain raw articulatory information that, however, is limited to basic behaviors of the vocal tract. On the contrary, acoustic features incorporate complex information, including the co-articulation effects. The purpose is to embed raw articulatory and complex phenomena information to synthesize more accurate AFs. In order to do that, we examined three deep learning architectures: two of them are based on Autoencoders (AE1 and AE2), while the latter employs deep residual network.
The model offering the most accurate AFs is AE2, in which the LFs/SFs are provided as input and the network is trained to minimize their reconstruction. Simultaneously, AE2 is optimized to also minimize the difference, in norm, between the latent representation (i.e. the encoding) and the acoustic vector. In this context, the SFs turned out to outperform the LFs. Also in this case, we evaluated the reconstruction performance in both matched and mismatched training-testing condition. Surprisingly, the performances obtained using the SFs computed from the whole articulatory dataset were not substantially deteriorated when using only SF1/SF2, in which only one/two speakers data have been used for the statistical procedure. This striking result suggests

that the statistical features are a very effective tool to capture the average configuration of the vocal tract, even when very limited motor measurements are available.

2.3 Integration of Speech Production knowledge to ASR

In this section, we assume to have access to a small acoustic-articulatory dataset (D1) that we leverage to train an articulatory reconstruction mapping. Then, we consider a large audio-only dataset (D2) to perform Automatic Speech Recognition. Hence, here we confront the mismatched training-testing case. Specifically, we compute the statistical features on D1 and learn a mapping to generate the VTVs in a supervised framework as shown in Sec. 2.2. We then use the learned mapping to synthesize the AFs for D2 as showed in Fig. 2.12. This procedure is described in detail in Sec. 2.3.1. Finally, we investigate two possible approaches to integrate the motor estimations in the ASR (trained on D2) and enhance its performance. We expose them in Sec. 2.3.2 and 2.3.3. The experimental setup and the results are reported in Sec. 2.3.4 and 2.3.5, respectively.

2.3.1 AF synthesis

In Sec. 2.2.5, results showed that the best AF estimation for the mismatched condition is provided by the SFs-to-VTVs mapping. Therefore, here we only consider this model to learn the motor reconstruction function. In the following, we schematize the adopted procedure to synthesize the AFs:

1. We learn the SFS-to-VTVs mapping by using the XRMB dataset (for details, see Paragraph “Dataset” of Sec. 2.2.4). The network architecture and the training parameters are the ones reported in the Paragraph “Neural networks” of Sec. 2.2.4.

2. We compute the SFs of the training set of D2 and recover the VTVs by the learned map. We refer to the extracted motor gestures $\hat{\mathbf{z}}$ as Estimated VTVs (EVTVs).

3. We learn an acoustic-to-EVTVs mapping.

Let us remark that the first two steps require the use of the phone labels, that are supposed not to be available in the testing phase. As consequence, we can only compute the EVTVs of the training dataset. The third step is necessary if we want to have access to a VTV reconstruction mapping also for the testing phase. In practice, the mapping of step 3 is an acoustic inversion map where the standard target (i.e., the measured articulatory features) is substituted by an estimation of it (i.e., the EVTVs). This trick provides us with a reconstruction function based on acoustic features, that are available during both training and testing. We illustrate the proposed approach in comparison with the standard AI in Fig. 2.12.

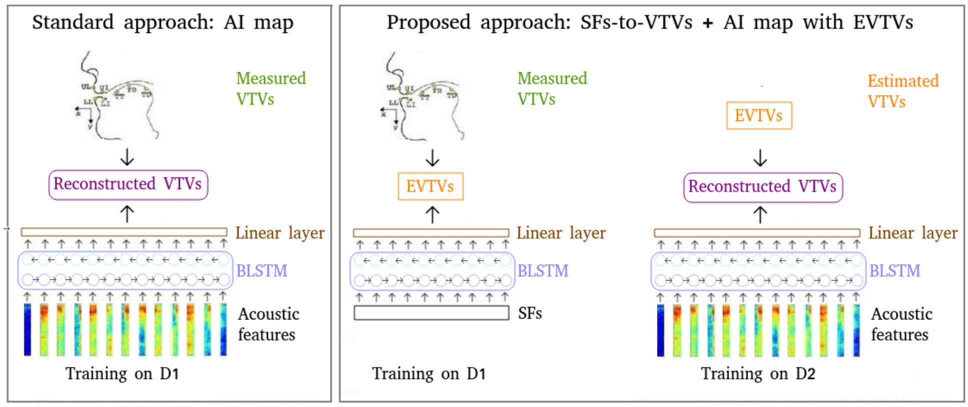


Figure 2.12: Comparison between the classical AF reconstruction method (i.e., the acoustic-inversion map) and the proposed one, based on Statistical Features.

2.3.2 ASR: articulatory as secondary target

The first strategy we investigate is the use of the speech production information as secondary target during the training of the speech recognizer. Note that, in this case, the AFs estimation is used only during training. Hence, we can directly employ the EVTVs without the need for the step 3 of Sec. 2.3.1.

We simultaneously learn the articulatory representation $\hat{\mathbf{z}}$ and the phonetic transcription $\mathbf{y}$. We denote the recovered AFs and phonemes with $\hat{\mathbf{y}}_{AF}$ and $\hat{\mathbf{y}}$, respectively. Therefore, we minimize the following objective function

$$L = L_{CE}(\mathbf{y}, \hat{\mathbf{y}}) + \lambda_{AF} \|\hat{\mathbf{y}}_{AF} - \hat{\mathbf{z}}\|_2^2 + \lambda_W \|W\|_2^2, \tag{2.5}$$

where L_{CE} is the cross-entropy function, λ_{AF} and λ_W are scalars weighting the articulatory loss and the L^2-norm weight regularization, respectively.

2.3.3 ASR: articulatory as additional input

We explore a second strategy in which the articulatory knowledge is used as additional input in the acoustic model. In this case, the AFs are required both during training and testing. Thus, we cannot use directly the EVTVs but we need to train the Acoustic-to-EVTVs mapping. Then, the output of the latter mapping is concatenated with the acoustic vector to provide the input of the acoustic model. Finally, this is learned by minimizing the cross-entropy function L_{CE}. Also in this case, we add a regularization term weighted by a scalar parameter $\lambda_W \|W\|_2^2$.

2.3.4 Experimental setup

Datasets We consider two well-known audio-only datasets: TIMIT [76] and CHiME-4 (noisy-clean). The TIMIT corpus consists of 2342 sentences read by 630 native speakers of American English with eight different dialects. Each reading contains ten phonetically rich sentences. The CHiME-4 database has been designed for the 4th edition of CHiME challenge and it is based on a vocabulary subset of the Wall Street Journal (WSJ0) corpus [77]. CHiME-4 consists of two type of data: Real data (recordings in a real noisy environments, such as a bus, a cafe, etc.) and Simulated data (noisy utterances artificially generated by mixing clean speech with noisy backgrounds). We use the same training-testing splitting proposed in the CHiME-4 challenge. The training set contains a total of 8738 noisy utterances (1600 real and 7138 simulated) from 83 speakers, while the testing set incorporates 2640 utterances (1320 real and 1320 simulated) recorded from other 4 speakers. Moreover, there are three types of recordings depending on the number of microphones available for testing (6-channel, 3-channel, 1-channel track) but we only consider the 1-channel tracks.

Both corpora are pre-processed and aligned by using Kaldi ASR toolkit [78]. Specifically, the audio data is preprocessed into 40 dimensional MFCCs, with velocity and acceleration, resulting in 120 dimensional vector. The frame size is 10 ms and the input window is 25 ms.

Neural networks

- **AFs as secondary target** The model architecture is a feedforward Deep Neural Network (DNN) with 4 layers, 2000 nodes par layer. Adaptive Momentum Optimizer is adopted with initial learning rate (LR) set to 0.1. Exponential decay is applied to the LR at each epoch. Table 2.7 resumes the chosen DNN parameters.

DNN Parameters	Value
Number of hidden layers	4
Number of neurons	2000
Window size	5
LR	0.1
Decay rate	0.75

Table 2.7: AFs as secondary target: DNN parameters.

- **AFs as additional input** We implement both a feedforward and a recurrent Neural Network (NN). More specifically, the feedforward NN has the same architecture and parameters reported in Tab. 2.7. Concerning the recurrent NN, we adopt a Bidirectional Long-Short Term Memory (BLSTM) with 4 hidden layers, each containing 250 memory blocks, as shown in Tab. 2.8. The optimization is carried out by Adaptive Momentum Optimizer, with the momentum parameter set to 0.9. The initial learning rate is 0.1 and it exponentially decays at every epoch.

BLSTM Parameters	**Value**
Number of hidden layers	4
Number of cells	250
Starting LR	0.1
Decay rate	0.75
Batch size	10

Table 2.8: AFs as additional input: BLSTM parameters.

In all the experiments, we initialize the DNN weights with Xavier initialization (*xavier*) or uniform distribution (*uniform*) between -0.5 and 0.5 rescaled by $\frac{2}{d}$ where d is the input dimension, while we used Xavier initialization for the BLSTM network.

2.3.5 Results

In the following we present the ASR performance with speech production knowledge. In particular, the classification task on the TIMIT dataset is the phoneme classification, while we perform senone classification on the CHiME-4 dataset. In both cases, we evaluate the Acoustic Model by the Frame Error Rate (FER). Further, for the CHiME-4 corpus, we incorporate the Acoustic Model into an ASR system by using Kaldi toolkit [78]. To build the ASR system, we use a 3-gram Language Model and we measure its performance in terms of Word Error Rate (WER).
For both datasets, we found that the optimal λ_W is 0.001 when the AFs are used as secondary target. Also, for this strategy, the best weight initialization is the *uniform* one and the batch size has been fixed to 200.

TIMIT The phoneme classification performance provided by the use of the estimated VTVs secondary target is reported in Table 2.9. As we can observe, the FER decreases for higher values of λ_{AF} confirming that the use of speech production knowledge provides a more accurate Acoustic Model. Note that the case $\lambda_{AF} = 0$ corresponds to the *Baseline* case, in which only the audio information is exploited.

λ_{AF}	**FER**
0	37.05
0.005	37.05
0.05	36.97
0.75	36.70
1.0	**36.52**

Table 2.9: Evaluation on TIMIT of the Acoustic Model in which the estimated VTVs are used as secondary target.

Similar conclusions can be drawn from Table 2.10, in which we show the DNN and BLSTM performance when the acoustic representation is used alone or beside the articulatory information. We only report the DNN performance with the optimal weight initialization, that is the uniform one, and batch size equal to 200. For a better reading, in Table 2.10 we simply write AFs to refer to the synthesized VTVs (from the 3-steps procedure described Sec. 2.3.1). The motor information provides a better performance for both neural network architectures, even though for the BLSTM the improvement is less remarkable.

Network	**Input**	λ_W	**FER**
DNN	MFCCs	0.001	37.05
	MFCCs + AFs	**0.001**	**34.88**
BLSTM	MFCCs	0	33.89
	MFCCs + AFs	**0**	**33.82**

Table 2.10: Acoustic Model evaluation on TIMIT. We report the DNN and BLSTM performance in which the synthesized AFs are used as additional input to the acoustic one, and we compare it with the baseline case where only acoustic information are used.

CHiME-4 In the following, we denote with *real* the subset of real data, while *all* will refer to the whole CHiME-4 dataset (both real and simulated recordings).
We first analyze the advantages of the first strategy, in which the articulatory contribution derives from the secondary loss. Table 2.11 confirms the results obtained on the TIMIT data, showing a decreasing FER for increasing values of λ_{AF}. The lowest FER is achieved for $\lambda_{AF} = 1.0$. For this value and for $\lambda_{AF} = 0$ (corresponding to the *Baseline*), we also report the WER on the subset of Real recordings. The use of EVTVs as secondary target provides the best WER, thus improving the ASR performance.

λ_{AF}	**FER(*all*)**	**WER(*real*)**
0	59.04	24.02
0.005	58.94	-
0.05	59.02	-
0.75	58.78	-
1.0	**58.67**	**23.57**
1.5	58.75	-
2.0	58.70	-

Table 2.11: Acoustic Model and ASR performance on CHiME-4 when AFs are used as secondary target. We report the FER and WER at varying of λ_{AF} that weights the articulatory loss in Eq. 2.5.

In Table 2.12 we compare the recognition performance (in terms of both FER and WER) when the input network is given either by the acoustic sequence or by the concatenation of acoustic and articulatory information. Although we experimented two weight initializations, different batch size ($\{100, 200, \ldots, 1000\}$) and λ_W (from

0 to 0.01) parameters for the DNN, we here report only the first two best performances that have been achieved by uniform initialization and batch size= 1000. For the BLSTM, we did not investigate these hyperparameters but we fixed them as described in Sec. 2.3.4.

The lowest WER on is provided by the DNN in which the synthetic AFs are concatenated with the acoustic input. Moreover, the DNN with side information outperforms its counterpart with audio-only features also in terms of FER, both when tested on the whole dataset and the subset of the real data.
While the motor representation improves the ASR performance based on DNN, this does not apply to the BLSTM model when tested on the real data. A wider exploration of the hyperparameter space may improve the BLSTM performance. However, it is crucial to remark that the BLSTM underperforms the DNN in terms of WER. This is in contrast with the literature [79, 80]. Recently studies carried out by a co-author of this work suggested that adding deltas and delta-deltas to the 40-dimensional MFCCs deteriorates the performance of the BLSTM network. Therefore, it is reasonable to think that the extracted acoustic features are optimal for the DNN but suboptimal for the BLSTM. For this reason, we believe it is not worthwhile to discuss the BLSTM results of Table 2.12.

Network	**Input**	λ_W	**FER(*all*)**	**FER(*real*)**	**WER(*real*)**
DNN	MFCCs	0	61.18	63.6	22.77
	MFCCs	0.001	61.69	63.7	25.32
	MFCCs + AFs	**0**	**55.60**	**57.3**	**21.54**
	MFCCs + AFs	0.001	56.99	58.8	25.76
BLSTM	**MFCCs**	**0**	56.30	**52.9**	**25.67**
	MFCCs + AFs	0	**53.79**	53.0	26.32

Table 2.12: Acoustic Model and ASR performance on CHiME-4 dataset.

Outcomes on both CHiME-4 and TIMIT emphasize that integrating the articulatory knowledge always improves the phoneme and senone recognition, as it provides the best performance in all the proposed strategies. Further, on the real data CHiME-4 subset, the synthesized AFs also reduce the WER of 2% and 5% when used as secondary target or additional input, respectively. In all cases, we found that the best results are achieved by the strategy in which the motor information are concatenated with the acoustic input. This suggests that adding the intermediate step in the AFs synthetization procedure (described in Sec. 2.3.1) does not imply any loss of information.

2.4 Discussion and application on dysarthric speech

As suggested by the Articulatory Phonology theory [45, 46] and proved in several works [22–24, 63–65], the Acoustic-Articulatory ASR outperforms the standard ASR system. However, the majority of speech corpora does not have access to articulatory measurements. To address this issue, we focused on methods to synthesize AFs for audio-only datasets. In order to do that, we introduced the Statistical Features (SFs), representing the average configuration of the Vocal tract, and we proposed their use to generate a richer articulatory representation.

The main advantage of this type of features is that they do not depend on the recording conditions (as the acoustic ones) and, thus, are more robust to the variability across datasets. We investigated their use in a supervised framework and, specifically, in addition or substitution to the acoustic vector in the AI map, in order to improve its generalization to new datasets. Secondly, we explored unsupervised methods that aim at extracting raw articulatory information from the SFs and capturing the coarticulation effects from the acoustic features, in order to synthesize more accurate AFs. We refer to the AFs generated by these approaches as Estimated Vocal Tract Variables (EVTVs).

All the described methods rely on the use of phone labels that, however, are available only during training in ASR tasks. We offered two possible solutions. Firstly, we employed the EVTVs in ASR as secondary target in order to be required only during the training phase. Secondly, we proposed to learn an additional mapping in which the acoustic features are used as input and the target is given by the EVTVs. This allow to synthesize the AFs during both training and testing phase. Consequently, the generating AFs can be used in addition to the acoustic input in the ASR model.

Results showed that both strategies outperform the ASR model in which only acoustic information are used.

By directly using the EVTVs, we obtained a relative FER reduction of 1.4% and 0.6% on TIMIT and CHiME-4, respectively. For the latter corpus, this strategy provided a 23.57 WER while the WER is 24.02 in absence of articulatory information.

We found that the second approach, in which the synthesized AFs are concatenated with the acoustic input vector, outperforms the first strategy on both corpora by suggesting that the additional step, based on the acoustic-to-EVTVs mapping, does not deteriorate the articulatory representation.
We here implemented both a feedforward and a recurrent NN. Results showed that the DNN always outperforms the BLSTM, by contrasting the majority of the studies in literature [79, 80]. As aforementioned, we believe this discordant finding is due to the use of deltas and delta-deltas in the MFCC representation that deteriorate the BLSTM training. Therefore, we do not further discuss the BLSTM results and we focus instead on the ones provided by the DNN. We obtained a FER relative reduction of 5.9% and 9.1% on the TIMIT and CHiME-4 corpus, respectively, and a WER relative reduction of 5.4% on the real data of CHiME-4.

We strongly believe in the transferability of this work to ASR systems for dysarthric speech. Indeed, a tricky issue in dysarthric speech recognition is the mispronunciation

of some phonemes resulting in a low recognition performance. One of the first observed impairments of dysarthric people is the imprecise production of stop consonants such as /p/, /t/, /k/, /b/, /d/, and /g/ [81]. A reduction of the articulatory precision in stop consonants has also been observed in [82]. As the speech production knowledge turned out to enhance the phoneme and senone recognition, we believe the use of SFs can be exploited for the recognition of some consonants (e.g., to fine-tune the model in order to avoid the misrecognition).
Also, dysarthric speaker adaptation is a required step for the ASR improvement. However, the speaker data is typically very limited and cannot represent all the labels. For example, in senone recognition models the number of labels is very high. This is due to the fact that senones are context-dependent subwords that aim at better describing surface pronunciations. Dependencies between surface feature values can be encoded in smoothness constraints in the motion of articulators. Therefore, articulatory features offer a compact information able to cover all the phonetic representations.

Last but not least, we want to remark that having access to corpora containing both articulatory and dysarthric speech is very rare. Our approach provides a method to synthesize AFs when only the audio signals are available. Although the proposed SFs represent the canonical configuration of the vocal tract of a (healthy) speaker, the methods introduced in this Chapter allow to embed speaker-dependent information from the acoustic features and synthesize accurate AFs. Thus, this work may offer an efficient way to generate AFs for dysarthric speech corpora. As well as the aforementioned applications, the synthesized AFs may also be employed for clinical applications, including speech therapy. Indeed, as also shown in [83, 84], a visual articulatory feedback can be a powerful tool for speech rehabilitation and phonetic correction. This might be particularly effective in the treatment of some diseases affecting children that do not involve cognitive degeneration, such as childhood apraxia of speech.

2.5 Appendix: Linguistic and Statistical features tables

In this supplementary material, we report the look-up tables related to the Linguistic Features (LFs) and Statistical Features (SFs) introduced in Sec. 2.2.1.

Table 2.13 reports the estimation of the VTVs for each phoneme, based on linguistic observations. Some phonemes have been split into two sub-units (denoted by numbers 1 and 2). The separation into two parts is due to the fact that some LF values can change within the same phoneme (e.g., diphthongs). The temporal division is not necessarily even, i.e. the duration of the first and second sub-phonemes can be different. Let us assemble three groups of phonemes based on the sub-units duration.

- AW , AY , EY , OW , OY: $\frac{2}{3}$ of the overall frame to state 1, the remaining $\frac{1}{3}$ to state 2;
- B , D , G , P , T: half to state 1 and half to state 2;
- CH , GH: the first $\frac{1}{3}$ of overall frame to state 1, the last $\frac{2}{3}$ to state 2.

As we can see, the features in Table 2.13 lie within a fixed discrete range. The chosen interval is not equal for all motor features. On average, there are 5 quantization levels per feature.

Phoneme	**LP**	**LA**	**TTCL**	**TTCD**	**TBCL**	**TBCD**	**VEL**	**GLO**	**Consonant**	**Silence**
AA	1	3	1	5	3	3	0	1	0	0
AE	1	3	1	5	1	5	0	1	0	0
AH	1	3	1	4	2	4	0	1	0	0
AO	0	3	1	5	3	3	0	1	0	0
AW1	1	3	1	5	1	5	0	1	0	0
AW2	0	2	2	5	2	3	0	1	0	0
AY1	1	3	1	5	3	3	0	1	0	0
AY2	1	3	1	3	0	3	0	1	0	0
B1	1	0	1	4	2	5	0	1	1	0
B2	1	1	1	2	2	5	0	1	1	0
CH1	1	3	1	0	1	4	0	2	1	0
CH2	1	3	2	1	0	3	0	2	1	0
D1	1	3	1	1	1	4	0	1	1	0
D2	1	3	1	0	1	4	0	1	1	0
DH	1	3	0	1	2	4	0	1	1	0
EH	1	3	1	4	0	4	0	1	0	0
ER	1	3	3	2	2	5	0	1	0	0
EY1	1	3	1	4	0	4	0	1	1	0
EY2	1	3	1	3	0	3	0	1	1	0
F	2	1	1	4	1	4	0	2	1	0
G1	1	3	2	5	1	0	0	1	1	0
G2	1	3	2	5	1	1	0	1	1	0
HH	1	3	1	4	2	4	0	2	1	0
IH	1	3	1	3	0	3	0	1	0	0
IY	1	3	1	3	0	2	0	1	0	0
JH1	1	3	1	0	1	4	0	1	1	0
JH2	1	3	2	1	0	4	0	1	1	0
K1	1	3	2	5	1	0	0	2	1	0
K2	1	3	2	5	1	1	0	2	1	0
L	1	3	1	0	2	2	0	1	1	0
M	1	0	1	4	2	4	1	1	1	0
N	1	3	1	0	2	4	1	1	1	0
NG	1	3	2	5	1	0	1	1	1	0
OW1	0	3	2	5	2	3	0	1	0	0
OW2	0	2	2	5	1	2	0	1	0	0
OY1	0	3	1	5	2	3	0	1	0	0
OY2	1	3	1	3	0	3	0	1	0	0
P1	1	0	1	4	2	5	0	2	1	0
P2	1	1	1	4	2	5	0	2	1	0
R	1	3	3	2	2	5	0	1	1	0
S	1	3	1	1	2	4	0	2	1	0
SH	1	3	2	1	0	3	0	2	1	0
T1	1	3	1	0	1	4	0	2	1	0
T2	1	3	1	1	1	4	0	2	1	0
TH	1	3	0	1	2	4	0	2	1	0
UH	0	3	2	5	2	3	0	1	0	0
UW	0	2	2	5	1	2	0	1	0	0
V	2	1	1	4	1	4	0	1	1	0
W	0	2	2	5	2	2	0	1	1	0
Y	1	3	1	3	0	2	0	1	1	0
Z	1	3	1	1	2	4	0	1	1	0
ZH	1	3	2	1	0	4	0	1	1	0
REST1	1	0	1	0	1	3	0	0	0	1
REST2	1	0	1	0	1	3	0	0	0	1

Table 2.13: Look-up table for linguist features.

Phoneme	LP	LA	TTCL	TTCD	TBCL	TBCD	VEL	GLO	Consonant	Silence
AA	0	1	1	1	1	1	0	1	0	0
AE	1	1	0	1	0	0	0	1	0	0
AH	0	0	0	0	0	0	0	1	0	0
AO	0	0	1	1	1	0	0	1	0	0
AW	0	0	1	1	1	0	0	1	0	0
AY	1	1	0	1	0	0	0	1	0	0
B	-1	-1	0	1	1	0	0	1	1	0
CH	-1	0	1	-1	0	0	0	2	1	0
D	0	0	0	-1	-1	1	0	1	1	0
DH	0	0	-1	-1	-1	0	0	1	1	0
EH	1	1	0	0	0	1	0	1	0	0
ER	0	0	1	0	0	1	0	1	0	0
EY	1	1	0	0	-1	0	0	1	1	0
F	0	-1	1	1	1	0	0	2	1	0
G	0	0	0	0	0	-2	0	1	1	0
HH	0	1	0	0	0	0	0	2	1	0
IH	0	0	0	0	-1	0	0	1	0	0
IY	0	0	-1	-1	-1	0	0	1	0	0
JH	-1	0	1	-1	0	1	0	1	1	0
K	0	0	0	0	0	-2	0	2	1	0
L	0	0	0	0	1	0	0	1	1	0
M	-1	-2	0	1	0	0	1	1	1	0
N	0	0	0	-1	-1	1	1	1	1	0
NG	1	0	0	0	0	-2	1	1	1	0
OW	-1	0	1	2	2	-1	0	1	0	0
OY	0	0	1	1	1	0	0	1	0	0
P	-1	-1	1	1	1	0	0	2	1	0
R	0	0	1	0	0	1	0	1	1	0
S	0	0	0	-1	0	0	0	2	1	0
SH	-1	0	1	-1	0	0	0	2	1	0
T	0	0	0	-1	0	0	0	2	1	0
TH	0	0	-1	-1	-1	0	0	2	1	0
UH	0	0	0	0	0	0	0	1	0	0
UW	-1	-1	0	0	0	-1	0	1	0	0
V	0	-1	0	0	0	0	0	1	1	0
W	-1	-1	1	2	2	-1	0	1	1	0
Y	0	0	-1	-1	-1	0	0	1	1	0
Z	0	0	0	-1	0	0	0	1	1	0
ZH	-1	0	1	-1	0	1	0	1	1	0
REST1	-1	-2	-1	-1	-1	2	0	0	0	1
REST2	0	0	0	0	0	0	0	0	0	1

Table 2.14: Look-up table for statistical features.

Chapter 3

Multi-sources domain adaptation

Traditional ASR systems usually require a large amount of data to well generalize to a new speaker or dataset. Even thought we have access to several large healthy speech corpora, all the existing dysarthric speech datasets are very limited. This is due to the difficulties with obtaining the healthcare service's approval for the speech recordings and with the issues related to the patient's disease (e.g., the fatigue during the recording session). A successful strategy to proceed when the dataset of interest is too small is training the model on a similar, but larger, dataset and then fine-tuning it on the dataset of interest. This procedure is known in machine learning as *domain adaptation*. In the case in which the dataset of interest consists of audio signal recorded from dysarthric speakers, we can train the system on one or more datasets that can contain both healthy or dysarthric speech. The same procedure can also be adopted when the dataset of interest contains recordings from only one speaker. In this case, we refer to it as *speaker adaptation*.

In this Chapter, we propose a method for domain adaptation (DA) when multiple datasets are available with the final purpose of confronting the problem in dysarthric speech. More precisely, in Sec. 3.1, we introduce the DA problem and the particular case of speaker adaptation. Sec. 3.2 recalls the Optimal Transport (OT) Theory and an OT-based DA method, named Joint Distribution Optimal Transport (JDOT), firstly introduced in [85] that is crucial for the understanding of the following sections. We then propose a new DA approach, that we call Multi-Source Domain Adaptation via Weighted Joint Distribution Optimal Transport (MSDA-WJDOT), in Sec. 3.3 where we also provide theoretical findings. To analyze the efficiency and robustness of MSDA-WJDOT algorithm, we conduct a preliminary study in Sec. 3.4 in which we also test the method on simulated data. Finally, we evaluate it on real data and, in particular, on dysarthric speech (Sec. 3.5).

3.1 Domain adaptation

Many machine learning algorithms assume that the test and training datasets are sampled from the same distribution. However, in many real-world applications, new data can exhibit a distribution change (*domain shift*) that degrades the algorithm performance. This shift can be observed for instance in speech recognition when the recording conditions or speaker accents are varying, or on computer vision when changing background, location, illumination or pose of the test images. This problem, known as Domain Adaptation (DA) [26, 27], is a particular case of transfer learning [28]. The population of interest (e.g., the test dataset) is called *target* domain, for which the labels are usually assumed not available. DA methods leverage a similar labelled dataset, called *source* domain, in order to learn a classifier for the target domain.

Several DA methods incorporate a distribution discrepancy loss into a neural network to reduce the domain gap. The distances between distributions are usually measured through an adversarial loss [86–89] or integral probability metrics, such as the maximum mean discrepancy [90, 91]. Recently, DA techniques based on Optimal Transport have been proposed in [85, 92, 93] and justified theoretically in [94].

In this Chapter, we focus on the setting, more common in practice, in which several labelled sources are available and we aim at transferring knowledge to an unlabelled target domain. In the following, we refer to the addressed problem as multi-source domain adaptation (MSDA) problem.

Many recent approaches motivated by theoretical considerations have been proposed for this problem. For instance, [95, 96] provided theoretical guarantees on how several source predictors can be combined using proxy measures, such as the accuracy of a hypothesis. This approach can achieve a low error predictor on the target domain, under the assumption that the target distribution can be written as a convex combination of the source distributions.
Other recent methods [97–99] look for a single hypothesis that minimizes the convex combination of its error on all source domains and they provide theoretical bounds of the error of the obtained hypothesis on the target domain. These guarantees generally involve some terms depending on the distance between each source distribution and the target distribution and suggest to find an embedding in which the feature distributions between sources and target are as close as possible, by using Adversarial Learning [98, 100, 101] or Moment Matching [97].

However, it may not be possible to find an embedding preserving discrimination even when the distances between source and target marginals are small. For example, when the sources are obtained by a rotation, the existence of such invariant embedding is prevented as theorized in [102].

3.1.1 Speaker adaptation

As mentioned above, the problem of domain shift between training and testing is also present in ASR [103]. Beside the recording conditions mismatching, ASR suffers of differences between speakers. Specifically, different speakers can show different vocal tracts and different accents. They can also have different speaking styles (e.g., speaking rate). This mismatch is even more evident in dysarthric people as the speech characteristics also depend on the type and the severity of dysarthria. This problem is known as speaker adaptation.

More specifically, let us suppose to have access to a speaker-independent (SI) acoustic model that provides good performances on the speech of all speakers in general. The purpose of speaker adaptation is to adapt the acoustic model to the target speaker in order to achieve the optimal ASR performance on her speech, comparable with speaker-dependent (SD) acoustic models performance. Typically, only a very limited amount of target speech data are available and, at the same time, SI models are usually based on deep neural networks (DNNs) with a large number of parameters. As a consequence, the model can easily start to overfit. This makes the speaker adaptation task very challenging.

To overcome this problem, many approaches aim at reducing the number of parameters to adapt. [104, 105] propose to insert a linear layer in the SI model and train it instead of re-weighting the whole SI model. In [106–108], the authors reduce the trainable parameters by singular value decomposition (SVD) of the neural network weight matrices. Other methods introduced auxiliary speaker information in addition to the speech. In [31], the DNN performance is improved by concatenating each frame with speaker identity vectors (i-vectors), while speaker-code are used in [107, 109]. In [110, 111], the speaker information in input is represented by the feature space Maximum Likelihood Linear Regression (fMLLR) transformed features, i.e. acoustic features transformed to speaker adapted features in order to maximize the likelihood of the adaptation data.

More recently, the advent of adversarial learning reached both domain and speaker adaptation in ASR. [112] investigates the adversarial multi-task learning framework to address the unsupervised adaptation. In this work, two discriminative classifiers are jointly learned sharing the same DNN layers. The main task is the phoneme classification, while the secondary task discriminates between the source and the target domains. The model is optimized by minimizing the loss of the main task and maximizing the discrimination loss.
In [113], adversarial teacher-student learning is adopted for condition-robust unsupervised domain adaptation. In this method, a student acoustic model and a condition classifier are jointly trained by 1) minimizing the Kullback-Leibler (KL) divergence between the output distributions of the teacher and student models; 2) min-maximizing the condition classification loss. To achieve condition-robustness, a condition-invariant and senone-discriminative deep feature are learned in the adapted student model through this procedure.
The authors in [114] propose an adversarial multi-task learning (MTL) approach to regularize the distribution of the hidden representations in a Speaker-Dependent (SD) DNN model to make it “close” to the Speaker-Independent (SI) one. A discriminator is trained to distinguish between the hidden representations generated by the SI and SD model. The latter one is optimized by, simultaneously, maximizing the SI/SD discrimination loss and minimizing the senone classification loss. A similar approach is proposed in [115], where a regularized adaptation technique is proposed for context

dependent DNN Hidden Markov Models (CD-DNN-HMM). In this paper, the distribution of the SD model is forced to be close to the one of the SI model by minimizing the KL-divergence. Other regularization-based techniques for adaptation have been also introduced in [116, 117].

To the best of our knowledge, none of the previous DA methods for ASR has been based on optimal transport.

3.2 Previous works

In this section we first recall the Optimal Transport problem and the notion of Wasserstein distance. Then we discuss how they were exploited for domain adaptation (DA) in the Joint Distribution Optimal Transport (JDOT) formulation that will be central in our approach.

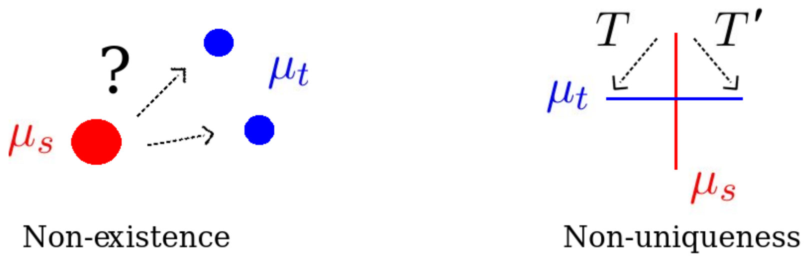
Notations Let $g : \mathcal{X} \to \mathcal{G}$ be a differentiable embedding function, with $\mathcal{G}$ the embedding space. Through the paper all input distributions are in this embedding space. We let p_S be the true distribution in the source domain and p_T the true distribution in the target, both supported on the product space $\mathcal{G} \times \mathcal{Y}$, where $\mathcal{Y}$ is the label space. In practice we only have access to a finite number N_S of samples in the source domain leading to the empirical source distribution $\hat{p}_S = \frac{1}{N_S} \sum_{i=1}^{N_S} \delta_{g(x_S^i), y_S^i}$ where δ is the Dirac function. In the target domain we only have access to a finite number of unlabelled samples N_T in the feature space and to $\hat{\mu}_T = \frac{1}{N_T} \sum_{i=1}^{N_T} \delta_{g(x_T^i)}$, the empirical target marginal distribution. We denote with $T_{\#f} = g$ the push forward operator T such that $f(T^{-1}(x)) = g(x)$. Finally, given a loss function L and a joint distribution p, the expected loss of a function f is defined as $\varepsilon_p(f) = \mathbb{E}_{(x,y)\sim p}[L(y, f(x))]$.

3.2.1 Introduction to Optimal Transport

Monge formulation The Optimal transport (OT) problem has been originally introduced by Gaspard Monge in 1784 [32], with the objective of transporting and reshaping a pile of soil to form an embankment with minimal effort. In this context, the source distribution μ_S is the mass distribution of the pile of soil and the target distribution represents the one of the embankment. Monge aimed at finding a map T transporting μ_S into μ_T, optimal with respect a given cost function c that measures the effort required for moving sand from one point to another.

This problem can be formalized as follows. Let μ_S and μ_T be the source and target distributions and let then be given a cost function $c : \mathcal{G} \times \mathcal{G} \to \mathbb{R}_+$. The Monge problem is to find a transport map $T : \mathcal{G} \to \mathcal{G}$ satisfying $T_{\#\mu_S} = \mu_T$ such that T minimizes the cost functional

$$\int_{\mathcal{G}} c(x, T(x)) \mu_s(x) dx. \tag{3.1}$$



Non-existence

Non-uniqueness

Figure 3.1: (left) Example of a case in which a solution for the Monge problem does not exist. (right) Example of a case in which the solution for the Monge problem is not unique.

However, this optimization problem is an ill-posed problem as it is non convex and a solution T^* could not exist. This is for instance the case in which μ_S is a Dirac measure and μ_T is not (Fig. 3.1, left). Moreover, there is also no unicity in the solution of Eq. 3.1 (Fig. 3.1, right).

For these reasons, this problem remained unsolved for over 200 years, until some big mathematical breakthroughs in the 1980s and 1990s. Brenier [118] proved that when μ_S and μ_T have densities and the cost is the squared euclidean distance $c(x, x') = \|x - x'\|^2$, the Monge map T exists and is unique.

Kantorovich formulation In 1940, Kantorovich [119] proposed a relaxed formulation of the Monge problem. For the thesis purpose, we here report the formulation for discrete probability measures $\hat{\mu}_S = \sum_i a^i_S \delta_{x^i_S}$, $\hat{\mu}_T = \sum_i a^i_T \delta_{x^i_T}$, with $\sum_i a^i_k = 1$ and $a^i_k \geq 0, \forall i, k$. The Kantorovich OT problem searches a transport plan

$$\pi \in \Pi(\mu_S, \mu_T) := \{\pi \geq 0 | \sum_i \pi_{i,j} = a^j_T, \sum_j \pi i, j = a^i_S\},$$

i.e. the set of joint probabilities with marginals μ_1 and μ_2, that solves the following problem:

$$\min_{\pi \in \Pi(\mu_S, \mu_T)} \sum_{ij} C_{ij} \cdot \pi_{ij}. \tag{3.2}$$

$C_{ij} = c(x^i_S, x^j_T)$ represents the cost of transporting mass between x^i_S and x^j_T for a given ground cost function $c : \mathcal{G} \times \mathcal{G} \rightarrow \mathbb{R}_+$. Solving Eq. 3.2 is a linear program and it always have a solution if c is semi lower continuous.

Problem 3.2 can be expressed in its dual formulation, given by the Rockafellar-Fenchel theorem, as follow

$$\max_{u,v \in C(\mathcal{G})} \left\{ \sum_i u(x^i_S) \mu_S(x^i_S) + \sum_j v(x^j_T) \mu_T(x^j_T) \right\} \tag{3.3}$$

with the constraint $u(x_S^i) + v(x_T^j) \leq c(x_S^i, x_T^j)$. The scalar functions u and v are called *Kantorovich* potentials and they are the dual variables of the optimization problem.

Wasserstein distance Eq. 3.2 also provides a measure of distance between the source and target. The Wasserstein distance between μ_S and μ_T is defined as

$$W_p(\mu_S, \mu_T) = \min_{\pi \in \Pi(\mu_S, \mu_T)} \left\{ \sum_{ij} C_{ij} \cdot \pi_{ij} \right\}^{\frac{1}{p}}, \tag{3.4}$$

where $C_{ij} = \|x_S^i - x_T^j\|^p$, and $p \geq 1$. $W_p(\mu_S, \mu_T)$ corresponds to the minimal cost for mapping one distribution to the other and $\pi^\star$ is the OT matrix describing the relations between source and target samples. C is often chosen to be the Euclidean distance, recovering the classical W_1 Wasserstein distance. The Wasserstein distance has been used with success in numerous machine learning applications such as Generative Adversarial Modeling [120, 121] and DA [85, 92, 122] thanks to its interesting properties. Indeed, contrary to the main divergences and measures of discrepancy between distributions (e.g., the Kullback–Leibler divergence), the Wasserstein distance provides a well-defined distance even when the distributions do not share the same support. Moreover, even though the Wasserstein distance is not differentiable, any solution u^* of the dual formulation 3.4 is a sub-gradient of W_d w.r.t. the source distribution weights $\boldsymbol{a_S}$, i.e.

$$\nabla_{\boldsymbol{a_S}} W_d(\mu_S, \mu_T) = u^*.$$

For more details, we refer the reader to Villani books [123, 124] and the book by Peyré and Cuturi [125] for the computational aspects of OT.

3.2.2 Joint Distribution Optimal Transport (JDOT)

This method has been proposed in [85] to address the problem of unsupervised DA with only one joint source distribution $\hat{p}_S$ and the feature marginal target distribution $\hat{\mu}_T$. Since no labels are available in the target domain, the authors proposed to use a proxy joint empirical distribution $\hat{p}_T^f$ whereby labels are replaced by the prediction of a classifier $f : \mathcal{G} \to \mathcal{Y}$, that is

$$\hat{p}_T^f = \frac{1}{N_T} \sum_{i=1}^{N_T} \delta_{g(x_T^i), f(g(x_T^i))}. \tag{3.5}$$

In order to use a joint distribution in the Wasserstein distance, they define, for $z, z' \in \mathcal{G}$ and $y, y' \in \mathcal{Y}$, the cost

$$D(z, y; z', y') = \beta \|z - z'\|^2 + L(y, y')$$

where L is a loss between classes and β weights the strength of feature loss. This cost takes into account embedding and label discrepancy. To train a meaningful classifier on the target domain, the authors of [85] solved the optimization problem

$$\min_{f} W_D(\hat{p}_S, \hat{p}_T^f) \tag{3.6}$$

where the objective function $W_D(\hat{p}_S, \hat{p}_T^f)$ is a Wasserstein distance between the joint source and joint "predicted" target

$$\min_{\pi \in \Pi(\hat{p}_S, \hat{p}_T^f)} \sum_{ij} D(g(x_S^i), y_S^i; g(x_T^j), f(g(x_T^j))) \cdot \pi_{ij}$$

and the minimization in (3.6) is over a suitable class of classifiers.

JDOT has been supported by generalization error guarantees, see [85] for a discussion. It was later extended to deep learning framework where the embedding g was estimated simultaneously with the classifier f with an efficient stochastic optimization procedure in [93]. One very important aspect of JDOT, that was overlooked by the domain adaptation community, is the fact that the optimization problem involves the joint embedding/label distribution. This is in contrast to a large majority of DA approaches [86, 122, 126] using divergences only on the marginal distributions, whereas using simultaneously feature and labels information is the basis of most generalization bounds as discussed in the next section.

3.3 Multi-Sources Adaptation via Weighted Joint Optimal Transport

In this section, we propose the Multi-Source Domain via Weighted Joint Optimal Transport (MSDA-WJDOT) method that approaches to the MSDA problem considering the diversity of sources distributions and taking advantage of this by selecting the sources closest to the target domain, in the Wasserstein sense.

More specifically, MSDA-WJDOT looks for a convex combination of the joint source distributions with minimal Wasserstein distance to an estimated target distribution, without relying on a proxy measure such as the accuracy of source predictors. We support this novel conceptual approach by deriving a generalization bound on the target error. Our algorithm consists in optimizing the term in this generalization bound, given by the Wasserstein distance between the estimated joint target distribution and a weighted sum of the joint source distributions.

One unique feature of our approach is that the weights of the source distribution are learned simultaneously with the classification function, which allows us to distribute the mass based on the similarity of the sources with the target, both in the feature and in the output spaces. As such, our model can also handle problems in which only target shift occurs. Interestingly the estimated weights provide a measure of domain relatedness and interpretability.

In the following, we assume to have J sources with joint distributions $p_{S,j}$, for $1 \leq j \leq J$. We define a convex combination of the source distributions

$$p_S^\alpha = \sum_{j=1}^{J} \alpha_j p_{S,j} \tag{3.7}$$

with $\boldsymbol{\alpha} \in \Delta^J$

and we present a novel generalization bound for MSDA problem that depends on p_S^α. Then, we introduce the MSDA-WJDOT optimization problem and propose an algorithm to solve it. Finally, we compare it with the state-of-the-art DA models.

3.3.1 Generalization Bound

The theoretical limits of domain adaptation are well studied and well understood since the work of [127] that provided an "impossibility theorem" showing that, if the target distribution is too different from the source distribution, adaptation is not possible. However in the case of MSDA, one can exploit the diversity of the source domains and use only the sources close to the target distribution, thereby obtaining a better generalization bound. For this purpose, a relevant assumption, already considered in [95], is to assume that the target distribution is a convex combination of the source distributions. The soundness of such an approach is illustrated in the following lemma.

Lemma 1. *For any hypothesis $f \in \mathcal{H}$, denote by $\varepsilon_{p_T}(f)$ and $\varepsilon_{p_S^\alpha}(f)$, the expected loss of f on the target distribution and on the weighted sum of the source distributions, with respect to a loss function L bounded by B. Then we have that*

$$\varepsilon_{p_T}(f) \leq \varepsilon_{p_S^{\boldsymbol{\alpha}}}(f) + B \cdot D_{TV}\left(p_S^{\boldsymbol{\alpha}}, p_T\right) \tag{3.8}$$

where D_{TV} is the total variation distance.

This simple inequality, whose proof is in the appendix, tells us that the key point for target generalization is to have a function f with low error on a combination of the joint source distribution and that combination should be "near" to the target distribution. Note that this also holds for single source DA problem corroborating the recent findings that just matching marginal distributions may not be sufficient [128].

While the above lemma provides a simple and principled guidance for a multi-source DA algorithm, it cannot be used for training since it assumes that labels in the target domain are known. In the following, we provide generalization bounds in a realistic scenario where no target labels are available and a self-labelling strategy is employed to compensate for the missing labels.

Taking inspiration from the result in Lemma 1, we propose a theoretically grounded framework for learning from multiple sources. To this end, we first recall the notion of Probabilistic Transfer Lipschitzness (PTL) of a classifier [85], that will be used in our method.

Definition 1. (PTL Property) *Let $\phi : \mathbb{R} \to [0, 1]$. A labeling function $f : \mathcal{G} \to \mathbb{R}$ and a joint distribution $\pi \in \Pi(\mu_S, \mu_T)$ are Φ-Lipschitz transferable if for all $\lambda > 0$, we have*

$$\text{Prob}_{(x_S, x_T) \sim \pi} \left[|f(x_S) - f(x_T)|| > \lambda D(x_S, x_T) \right] \leq \Phi(\lambda)$$

with D being a metric on $\mathcal{G}$

The PTL property is a reasonable assumption for DA that was introduced in [85] and provides a bound on the probability of finding pair of source-target samples of different label within a $1/\lambda$-ball.

Our approach is based on the idea that one can compensate the lack of target labels by using an hypothesis labelling function f which provides a joint distribution p_T^f in (3.5), where f is searched in order to align p_T^f with a weighted combination of source distributions p_S^{α}. Following this idea, we introduce the following generalization bound for MSDA.

Theorem 1. *Let $\mathcal{H}$ be a space of M-Lipschitz labelling functions. Assume that, for every $f \in \mathcal{H}$ and $x, x' \in \mathcal{G}$, $|f(x) - f(x')| \leq M$. Consider the following measure of similarity between p_S^{α} and p_T introduced in [127, Def. 5]*

$$\Lambda(p_S^{\alpha}, p_T) = \min_{f \in \mathcal{H}} \varepsilon_{p_S^{\alpha}}(f) + \varepsilon_{p_T}(f), \tag{3.9}$$

where the risk is measured w.r.t. to a symmetric and k-Lipschitz loss function that satisfies the triangle inequality. Further, assume that the minimizing function f^* satisfies the PTL property (Definition 1). Let $\hat{p}_{S,j}$ be j-th source empirical distributions of N_j samples and $\hat{p}_T$ the empirical target distribution with N_T samples. Then for all $\lambda > 0$, with $\beta = \lambda k$ in the ground metric D we have with probability at least $1 - \eta$ that

$$\begin{aligned} \varepsilon_{p_T}(f) \leq & W_D\left(\hat{p}_S^{\alpha}, \hat{p}_T^f\right) + \sqrt{\frac{2}{c'} \log \frac{2}{\eta}} \left(\frac{1}{N_T} + \sum_j \frac{\alpha_j}{N_j} \right) \\ & + \Lambda(p_S^{\alpha}, p_T) + kM\phi(\lambda). \end{aligned}$$

Note that the quantity $\Lambda(p_S^{\alpha}, p_T)$ in the bound measures the discrepancy between the true target distribution and the "best" combination of the source distributions. Interestingly the $1/N_j$ ratios in the bound are weighted by α_j which means that even if one source is poorly sampled it won't have a large impact as soon as the coefficient

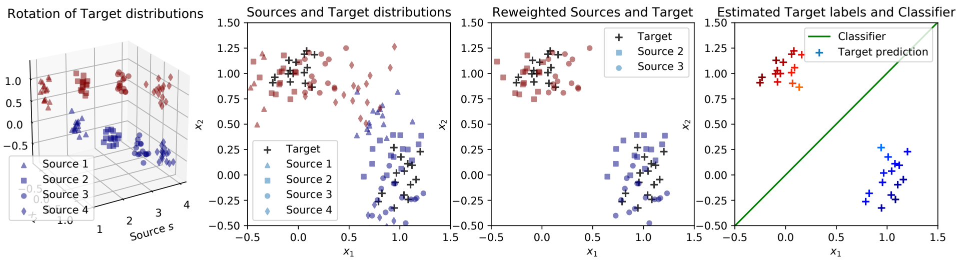


Figure 3.2: 2D simulated data. (left) illustration of 4 source distributions corresponding to 4 increasing rotations. The color of the sample corresponds to the class. (center left) source distributions and target distribution in black because no class information is available. (center right) source distributions weighted by the optimal $\boldsymbol{\alpha}^\star = [0, 0.5, 0.5, 0]$ from MSDA-WJDOT: only Source 2 and 3 have a weight > 0 because they are the closest to the target in the Wasserstein sense. (right) Final MSDA-WJDOT target classification.

α_j stays small. This suggests to investigate some kind of regularization for the weights $\boldsymbol{\alpha}$ but since it would introduce one more hyperparameter we left it to future studies and in the following we focus only on optimizing the first term of the bound.

The theorem above indicates that one can minimize the generalization error by optimizing both the predictor f and the weights $\boldsymbol{\alpha}$ of the source distributions. This is what we propose to do in the following.
The proof of the above enunciated Lemma and Theorem can be found in the Appendix 3.7.

3.3.2 MSDA-WJDOT method

Optimization Problem Our approach aims at finding a function f that aligns the distribution p_T^f with a convex combination $\sum_{j=1}^J \alpha_j p_{S,j}$ of the source distributions with convex weights $\boldsymbol{\alpha} \in \Delta^J$ on the simplex. We express the multi-domain adaptation problem as

$$\min_{\boldsymbol{\alpha}, f} \quad W_D\left(\hat{p}_T^f, \sum_{j=1}^J \alpha_j \hat{p}_{S,j}\right). \tag{3.10}$$

Problem above is a minimization of the first term in the bound from Theorem 1 with respect to both f and $\boldsymbol{\alpha}$. The role of the weight $\boldsymbol{\alpha}$ is crucial because it allows in practice to select (when $\boldsymbol{\alpha}$ is sparse) the source distributions that are the closest in the Wasserstein sense and use only those distributions to transfer label knowledge from.

An example of the method is provided in Figure 3.2 showing 4 source distributions in 2D obtained from rotation in the 2D space. One interesting property of our approach is that it can adapt to a lot of variability in the source distributions as long as the distributions lie in a distribution manifold and this manifold is sampled correctly by the source distributions. For instance the linear weights allow to interpolate between

Algorithm 1 Optimization for MSDA-WJDOT

```
Initialize $\boldsymbol{\alpha} = \frac{1}{J}\mathbf{1}_J$ and $\boldsymbol{\theta}$ parameters of $f_{\boldsymbol{\theta}}$ and steps $\mu_{\boldsymbol{\alpha}}$ and $\mu_{\boldsymbol{\theta}}$.
repeat
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \mu_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} W_D\left(\hat{p}_T^f, \sum_{j=1}^J \alpha_j \hat{p}_{S,j}\right)$
  $\boldsymbol{\alpha} \leftarrow P_{\Delta^J}\left(\boldsymbol{\alpha} - \mu_{\boldsymbol{\alpha}} \nabla_{\boldsymbol{\alpha}} W_D(\hat{p}_T^f, \sum_{j=1}^J \alpha_j \hat{p}_{S,j})\right)$
until Convergence
```

Algorithm 2 Projection to the simplex P_{Δ^J} [129]

```
Sort $\boldsymbol{w}$ into $\boldsymbol{u}$: $u_1 \geq \cdots \geq u_J$.
Set $K := \max_{1 \leq k \leq J}\{k|(\sum_{j=1}^k u_j - 1/k < u_k\}$.
Set $\tau := (\sum_{j=1}^K u_j - 1)/K$.
For $j = 1, \ldots, J$ set $\alpha_j := \max\{w_j - \tau, 0\}$.
```

source distributions and recover the weighted source that is the closest to the manifold of distribution, hence providing a tightest generalization as shown in the previous section.

Optimization Algorithm Problem (3.10) can be solved with a block coordinate descent similarly to what was proposed in [85]. But with the introduction of the weights $\boldsymbol{\alpha}$ we observed numerically that one can easily get stuck in a local minimum with poor performances. So we proposed the optimization approach in Algorithm 1, that is an alternated projected gradient descent *w.r.t.* the parameters $\boldsymbol{\theta}$ of the classifier $f_{\boldsymbol{\theta}}$ and the weights $\boldsymbol{\alpha}$ of the sources. Note that the sub-gradient of $\nabla_{\boldsymbol{\theta}} W$ is computed by solving the OT problem and using the fixed OT matrix to compute the gradient similarly to [93]. The sub-gradient $\nabla_{\boldsymbol{\alpha}} W$ can be computed in closed form from

$$\nabla_{\alpha_j} W_D\left(\sum_{j=1}^J \frac{\alpha_j}{N_j} \sum_{i=1}^{N_j} \delta_{(g(x_j^i),y_j^i)}, \hat{p}_T^f\right) = N_j \sum_{i=1}^{N_j} u_{j,i}^*$$

where $u_{j,i}^*$ is the dual variable for sample i in source domain j.

P_{Δ^J} is the projection to the simplex $\Delta^J = \{\boldsymbol{\alpha} \in \mathbb{R}^J | \sum_{j=1}^J \alpha_j = 1, \alpha_j \geq 0\}$ defined as

$$P_{\Delta^J}(\boldsymbol{w}) = \underset{\boldsymbol{\alpha} \in \Delta^J}{\operatorname{argmin}} \|\boldsymbol{w} - \boldsymbol{\alpha}\|. \tag{3.11}$$

We implemented it by using Algorithm 2, firstly proposed in [129].

Also note that Algorithm 1 can be performed on mini-batches by sub-sampling the source and target distribution on very large datasets as suggested in [93] which has been shown recently to provide robust estimators in [130].

Comparison with State-of-the-art MSDA-WJDOT is related to JDOT [85] but opens the door for a more general approach that can adapt to MSDA. There are two simple ways to apply JDOT to multi-source DA. The first one consists in concatenating all the source samples into one source distribution (equivalent to uniform $\boldsymbol{\alpha}$ if all N_j are equal) and using classical JDOT on the resulting distribution. The second one consists in optimizing a sum of JDOT losses for every source distribution but again, this leads to uniform impact of the sources on the estimation. It is clear that both approaches are not robust when some sources distributions are very different from the target (those would have a small weight in MSDA-WJDOT).

There exists a MSDA approach called JCPOT [131] based on [92] that has been proposed to address only target shift (change in proportions between the classes) and satisfies a generalization bound showing that estimating the class proportion in the target distribution is key to recovering good performances.
MSDA-WJDOT can also handle the target shift as a special case since the re-weighting $\boldsymbol{\alpha}$ is directly related to the proportion of classes. The main difference is that JCPOT estimates the proportions of classes using only the feature marginals, whereas MSDA-WJDOT estimates the proportion and classifier simultaneously by optimizing a Wasserstein distance in the joint embedding/label space. Also note that MSDA-WJDOT relies on a weighting of the samples where the weight is shared inside the source domains.

This is a similar approach to DA approaches such as Importance Weighted Empirical Risk Minimization (IWERM) [132] designed for Covariate Shift that use a reweighing of all the samples. One major difference is that we only estimate a relatively small number of weights in $\boldsymbol{\alpha}$ leading to a better posed statistical estimation. It is indeed well known that estimation of continuous density which is necessary for a proper individual reweighting of the samples is a very difficult problem in high dimension.

Finally, as discussed in the introduction, the majority of recent DA approaches based on deep learning [86, 122, 126] relies on the estimation of an embedding that is invariant to the domain which means that the final classifier is shared across all domains when the embedding g is estimated. Those approaches have been extended to multiple sources [97, 98, 100] with the objective that the embedded distributions between sources and target are similar.
Our approach differs greatly here for several reasons. First we do not try to cancel the variability across sources but to embrace it by allowing the approach to find the source domains closest in term of terms of embedding and classifier automatically. There exist numerous examples of source variability in real life (such as rotation between the full distributions) that cannot be handled with a global embedding and to the best of our knowledge MSDA-WJDOT is one of the few generic frameworks that can handle this problem.

3.4 Preliminary study

In this section, we first discuss the implementation and the robustness of MSDA-WJDOT. We then evaluate and compare it with state-of-the-art MSDA methods on simulated data. All the experiments have been carried out by using Python/Pytorch [133].

3.4.1 Practical implementation and algorithm robustness

Practical Implementation In all numerical experiments, we used the MSDA-WJDOT solver from Algorithm 1. We recall that we assume to have access to a meaningful (as in discriminant) embedding g. This is a realistic scenario due to the wide availability of pre-trained models and advent of reproducible research. Nevertheless we discuss here how to estimate such an embedding when none is available. To keep the variability of the sources that is used by MSDA-WJDOT we propose to estimate g with the Multi-Task Learning (MTL) framework originally proposed in [134], i.e.

$$\min_{g,\{f_j\}_{j=1}^{J}} \quad \sum_{j=1}^{J} \frac{1}{N_j} \sum_{i=1}^{N_j} \mathcal{L}(f_j \circ g(x_j^i), y_j^i). \tag{3.12}$$

This approach for estimating an embedding g makes sense because it promotes a g that is discriminant for all tasks but allows a variability thanks to the task-specific final classifiers f_j which is an assumption at the core of MSDA-WJDOT. We refer to MSDA-WJDOT where the embedding g is learned with the above procedure as `MSDA-WJDOT`$_{MTL}$ (and similarly, for `CJDOT`$_{MTL}$ and `MDJTO`$_{MTL}$).
When `MSDA-WJDOT` does not explicitly refer to the MTL, we learn g by

$$\min_{g,f_S} \quad \sum_{j=1}^{J} \frac{1}{N_j} \sum_{i=1}^{N_j} \mathcal{L}(f_S \circ g(x_j^i), y_j^i). \tag{3.13}$$

In this approach, that we will denote with `Baseline`, the classification function f_S is the same for all sources. Note that in both cases, this is a two step procedure in which we first learn g and then the target classifier through the MSDA-WJDOT algorithm.

An important question, especially when performing unsupervised DA, is how to perform the validation of the parameters including early stopping. We propose here to use the sum of squared errors (SSE) between the target points in the embedding and their cluster centroids. Specifically, we estimate cluster membership on the the outputs through $f \circ g$. Then the SSE is computed in the embedding g using the estimated clusters. Intuitively, if the SSE decreases it means that f attributes the same label to samples of the target domain that are close in the embedding.
A possible alternative strategy could be employing the accuracy of the learned classifier f on the source datasets and weighted by $\boldsymbol{\alpha}$, i.e.

$$\sum_{j=1}^{J} \alpha_j ACC_{S,j}(f), \tag{3.14}$$

with $ACC_{S,j}(f) = \frac{\#\{f(x_j^i)=y_j^i\}}{N_j}$. To refer to this approach, we denote as `MSDA-WJDOT`^{ACC}, `CJDOT`^{ACC}, `MJDOT`^{ACC} the MSDA-WJDOT and the two JDOT extensions respectively.

Let us remark that this is a way to reuse the weights $\boldsymbol{\alpha}$ that provide the closest source distributions which, hence, are supposed to give a better estimate of the performance of the current classifier.

In this work, we do not investigate the Lipschitz constant of f and suppose it being constant in the experiments. An alternative strategy could be penalizing an estimate of this constant, but this would add an extra parameter that requires validation parameter. We chose instead to limit the complexity of f with early stopping, which has the advantage of making optimization shorter.

Algorithm convergence and stability In Figure 3.3 (*Left* and *Center-left*) we show the stability of the algorithm for different weights initialization. The loss function always converges and the $\boldsymbol{\alpha}$ coefficients are not affected by the initialization. Moreover, we observed in practice that choosing the same step for $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$ does not degrade the performance and in all experiments we validated it via early stopping. We also noticed a fast convergence of the weights $\boldsymbol{\alpha}$, meaning that the relevant domains are quickly identified. This behavior is illustrated in Fig. 3.3 (*Right*), where $\boldsymbol{\alpha}$ sparsity rapidly increases for any choice of S illustrating that only a few relevant source distributions are used in practice. We also report the loss convergence for increasing number of sources S (*Center-right*).

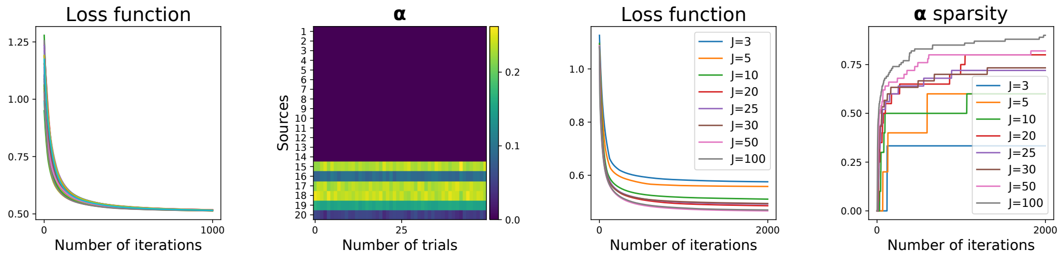


Figure 3.3: (left and center-left) Loss function and α coefficients with different weights initializations. (center-right and right) Loss function and α sparsity for increasing number of sources J.

3.4.2 Numerical experiments on Simulated Data

Compared Methods For the domain shift problem, we compare our approach with the following MSDA methods among which two non obvious extension of the JDOT formulation.

CJDOT consists in concatenating all the source samples into one source distribution.
MJDOT consists in optimizing the sum $\sum_j W_D(p_j, p_T^f)$ of JDOT objective for all sources. For both JDOT variants, we employ the SSE criterion discussed above to validate both parameters and early stopping.
Importance Weighted Empirical Risk Minimization (IWERM) [132] that is a variant of ERM where the samples are weighted by the ratio of the target and source densities minimizing the sum of the IWERM objective for each sources.
We also provide performances for Baseline that trains a classifier that maximizes

performance among all source domains. This approach measure the ability to train a unique classifier that is robust to all domains and performs well on target.
Finally, we also compare to two unrealistic approaches that use labels in target: `Baseline+Target` is similar to `Baseline` but also use labels in the target domain. `Target` trains a classifier using only target labels and is more prone to overfitting since less samples are available. Since we have access to labels for the two last approaches, we validate the model by using the classification accuracy on the target validation set making those two approaches clear upper bounds on the attainable performance for each dataset. All methods are compared on the same dataset split in training (70%), validation (20%) and testing (10%) but the validation set is used only for `Baseline+Target` and `Target`.
For the target shift problem, we compare the proposed method with `JCPOT`.

Domain Shift We consider a classification problem similar to what is illustrated in Figure 3.2, but with 3 classes (i.e. $\mathcal{Y} = \{0, 1, 2\}$) and in 3D. We generate a data set (X_0, Y_0) by drawing X_0 from a 3-dimensional Gaussian distribution with 3 cluster centers and standard deviation $\sigma = 0.8$. We keep the same number of examples for each cluster. To simulate the J sources, we apply J rotations to the input data X_0 around the x-axis. More precisely, we draw J equispaced angles θ_j from $[0, \frac{3}{2}\pi]$ and we get $X_j = \{\mathbf{x}_j^i\}$ as

$$\mathbf{x}_j^{i\top} = \mathbf{x}_0^{i\top} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\theta_j) & -sin(\theta_j) \\ 0 & sin(\theta_j) & cos(\theta_j) \end{bmatrix}. \tag{3.15}$$

To generate the target domain X_T, we follow the same procedure by randomly choosing an angle $\theta_T \in [0, \frac{3}{2}\pi]$. As the data is already linearly separated, we set g as the identity function and, hence, $\mathcal{X} \equiv \mathcal{G}$.

We carried out several experiments in order to study the effect of different parameters such as the number of source domains J, of source samples N_j and of target samples N_T. Each experiment has been repeated 50 times. We also investigated to replace the exact Wasserstein distance by the the Bures-Wasserstein distance

$$BW(\mu_S, \mu_T)^2 = \|\mathbf{m}_S - \mathbf{m}_T\|^2 + \text{Trace}\left(\Sigma_S + \Sigma_T - 2\left(\Sigma_S^{1/2}\Sigma_T\Sigma_S^{1/2}\right)^{1/2}\right), \tag{3.16}$$

where the $\mathbf{m}_S, \Sigma_S$ are respectively the first and second order moments of distribution μ_S (and similarly for $\mathbf{m}_T, \Sigma_T$). The BW distance has the advantage of having a complexity linear in the number of samples that can scale better to large dataset. We label this method variant with (B), while we refer to the exact OT as (E).

In the following, we investigate the performance of MSDA-WJDOT at varying of the number of sources J, source samples N_j, and target samples N_T, and we compare it with the `Baseline`, `Target`, `Bayes` classification, in addition to the MSDA State-of-the art methods performance.

- *Varying the number of sources:* we keep the number of samples fixed in both sources and target datasets (s.t. $N_j = N_T \ \forall j$) and we vary the number of sources $J \in \{3, 5, 10, 20, 25, 30\}$. In Fig. 3.4 we report the accuracy of the different methods.

- *Varying the number of source samples:* we fix the number of sources J equal to 20 and the number of target samples N_T to 300. Fig 3.5 shows the methods accuracy for varying the number of source samples N_j in $\{60, 180, 300\}$.

- *Varying the number of the target samples:* we fix $J = 20$ and $N_j = 300$, with $1 \leq j \leq J$. We let vary the number of target samples N_T in $\{60, 180, 300\}$ (Fig. 3.6).

- *Varying the number of samples of all domains*: we fix the number of sources equal to 20. We let vary the number of source and target samples in $\{60, 180, 300\}$, by keeping $N_j = N_T$ with $1 \leq j \leq J$. We report the methods' accuracy in Fig. 3.5.

In all experiments `MSDA-WJDOT` significantly outperforms all competitor methods both in term of performance and variance even for a limited number of sources or samples. Both `MSDA-WJDOT(E)` and `MSDA-WJDOT(B)` provide a comparable performance w.r.t. the `Target` method, in which the labels of the target dataset are used. Interestingly MSDA-WJDOT can even outperform `Target` due to its access to a larger number of samples.

Another important aspect of MSDA-WJDOT is the obtained weights $\boldsymbol{\alpha}$ that can be used for interpretation. In Fig. 3.8, 3.9 and 3.10 we show the recovered weights $\boldsymbol{\alpha}$ for a small number of sources ($J = 3$), small number of samples size ($N_j = N_T = 60$) and higher number of both sources and sample size ($J = 30$ and $N_j = N_T = 300$), respectively. In all cases, the x- axis reports different (sorted) random target angles in the $[0, \frac{3}{2}\pi]$ interval, whereas the y-axis represents the source angles ordered such that $\theta_j \leq \theta_{j+1}$, $1 \leq j \leq J - 1$. As we can see, the estimated weights tend to be sparse and put more mass along the diagonal meaning that `MSDA-WJDOT` always rewards the sources with angle closest to θ_T.

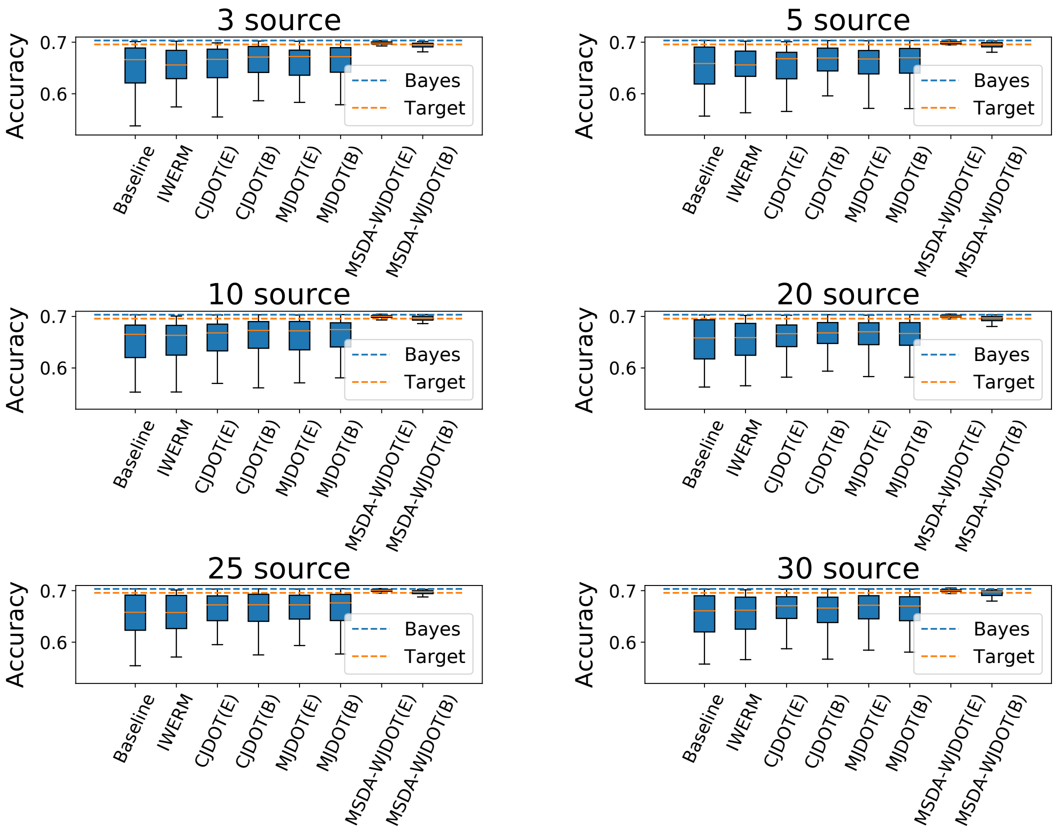


Figure 3.4: Methods' accuracy for varying the number of sources J.

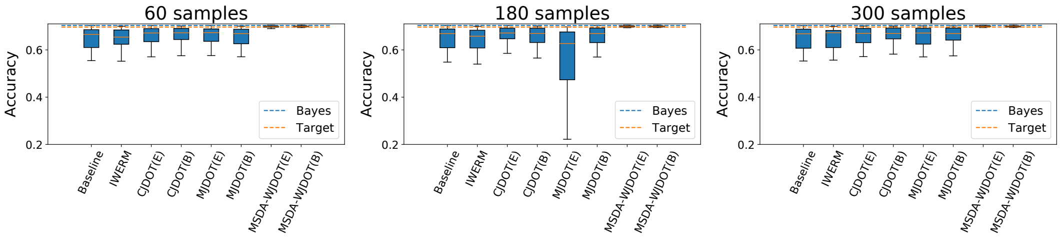


Figure 3.5: Methods' accuracy for varying the number of source samples.

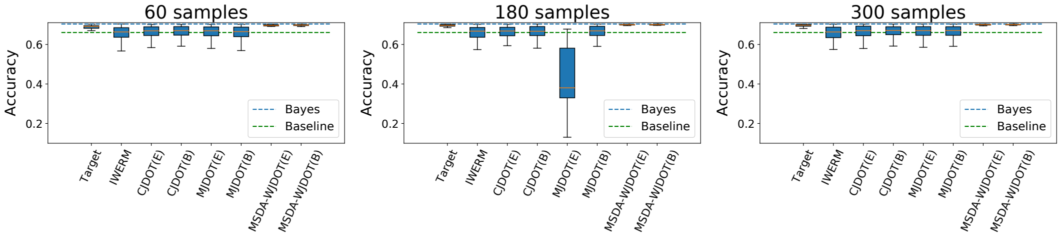


Figure 3.6: Methods' accuracy for varying the number of target samples

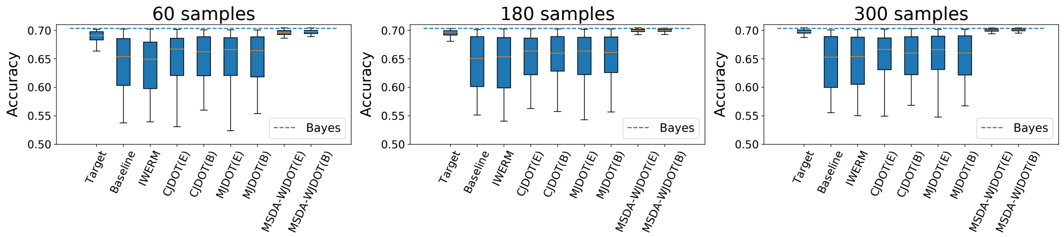


Figure 3.7: Methods' accuracy for varying the number of source and target samples

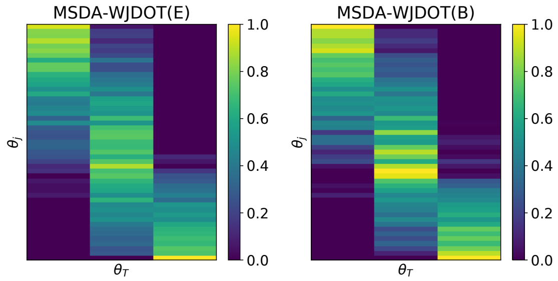


Figure 3.8: Recovered $\boldsymbol{\alpha}$ with small number of sources ($J = 3$ and $N_j = N_T = 300$).

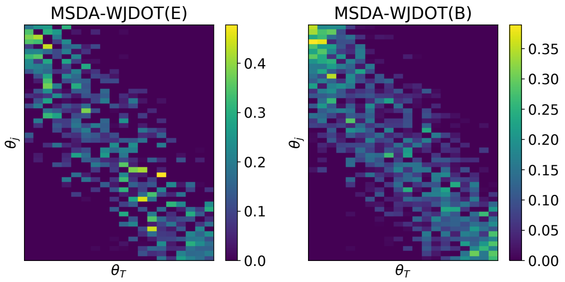


Figure 3.9: Recovered $\boldsymbol{\alpha}$ with small sample size ($J = 20$ and $N_j = N_T = 60$).

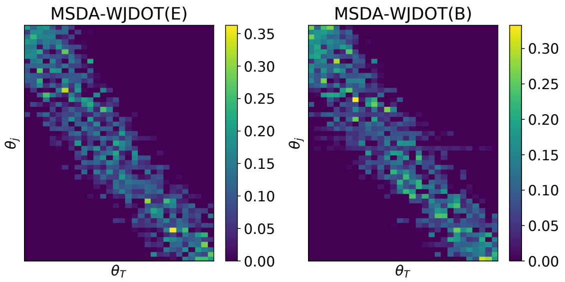


Figure 3.10: Recovered $\boldsymbol{\alpha}$ for $J = 30$ and $N_j = N_T = 300$.

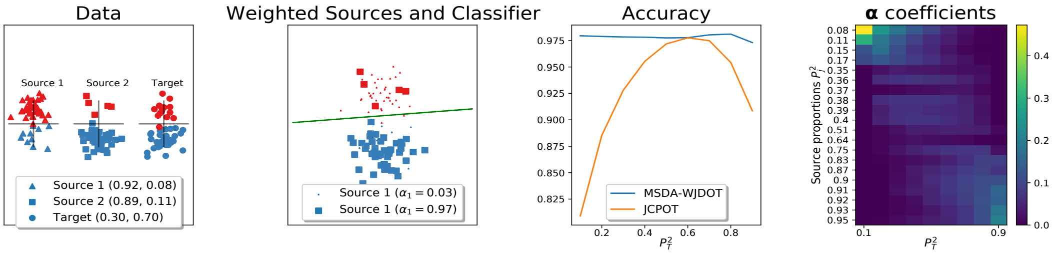


Figure 3.11: Illustration of MSDA-WJDOT on target shift problem. (left) illustration of 2 source and target distributions with unbalanced classes. (center) source distributions weighted by $\boldsymbol{\alpha}$ and estimated target classifier. (right) $\boldsymbol{\alpha}$ coefficients at varying of class proportions in target dataset.

Target Shift To investigate the potentiality of our method, we also take into account the target shift problem with 2D source and target datasets which presents different proportions of classes. The proportion of the class c in the source j is defined as

$$P_j^c = \frac{\#\{y_j^i = c\}}{N_j}$$

(and similarly for the target).

We consider a binary classification and we sample the sources and target samples from the same Gaussian distribution. In Fig. 3.11 (*Left and Center*) we illustrate 2 sources and target distributions and how MSDA-WJDOT reweights the sources. As we can see, almost all the mass is concentrated on Source 2 ($\alpha_2 \gg \alpha_1$) because its class proportion is closer to the target one. Instead, Source 1 has a class proportion inverted w.r.t. the target.

In the experiment reported in Fig. 3.11 (center-right and right) we have $J = 20$ sources with P_j^2 randomly generated between 0.1 and 0.9 (we ordered the sources s.t. $P_j^2 \leq P_{j+1}^2$). We show the average classification accuracy and the $\boldsymbol{\alpha}$ weights over 50 trials for varying P_T^2 in $\{0.1, 0.2, \cdots, 0.9\}$. Our method always outperforms JCPOT [131] and select the sources with a proportion of classes closer to the one in the target.

We did not further explore MSDA-WJDOT on target shift as it is beyond of the purpose of this thesis, but a future work could delve into this direction as many real-life problems present such an issue. For instance, dysarthric speech datasets are often non-homogeneous as, based on the dysarthria severity, the subject could be not able to record the ideal prefixed number of repetitions.

3.5 Applications to dysarthric speech

In this section, we deal with unsupervised domain adaptation and unsupervised speaker adaptation in pathological speech, when multiple sources are available.

3.5.1 Unsupervised Domain Adaptation for Dysarthria detection

We here focus on multi-source domain adaptation for the task of detecting dysarthria. Specifically, we assume to have access to multiple noisy datasets containing dysarthric and healthy speech. We employ MSDA-WJDOT to learn a binary classifier that performs dysarthria detection for an unlabelled noisy dataset.

Previous works Dysarthria detection and severity assessment are currently evaluated by neurology experts through the use of clinical assessment tools that attribute a score to the capacity of the subject to perform perceptual and/or acoustic tasks. The zero score corresponds to the healthy state, whereas scores higher than zero diagnose the presence of dysarthria.
Even though the role of the experts still remains fundamental, the detection might be biased and subjective. For example, the final score of dysarthria severity is strictly linked to the choice of the clinical assessment tool. Further, this procedure can be laborious and time consuming. Therefore, a rapid and objective dysarthria detection procedure could help the therapist in the diagnosis. It could also be employed to detect early signs of neurological disorder [135, 136] or to check the therapy progresses.

In the last years, the research community started to look at dysarthria detection by learning a mapping from the acoustic features to the text label [137, 138]. In [139], the authors consider an interpretable layer intermediate to the deep learning model. This layer acts as a bottle-neck feature extractor providing nasality, vocal quality, articulatory precision and prosody features. They showed that this interpretable features are highly correlated with the evaluation of Speech-Language Pathologists (SLPs). In [140] the authors propose a deep learning approach to compute phonological posteriors from the speech signal and model the voice quality spectrum in patient affected by Parkinson's Disease (PD). In [141], articulation impairments of PD patients are analyzed by time–frequency representations (TFRs). In particular, short time Fourier transform (STFT) is used for the onset in th speech recording (i.e., transition from voiced to unvoiced), and the wavelet transform is adopted for the offset (from unvoiced to voiced). Finally, a convolutional neural network (CNN) learns high–level representations from the low–level TFRs to discriminate between patients and healthy speakers.

Some studies have been conducted to detect dysarthria by prosodic measurements. In [142] eleven prosodic measurements (such as mean pitch, jitter, shimmer, articulation rate) are used in support vector machine (SVM) and Gaussian Mixture Model (GMM) classifiers to detect and assess dysarthria. In [143], seven rhythm metrics are provided in addition to the speech as input to GMM and multilayer perceptron (MLP) classifiers. Recently, the authors in [144] focused on rhythm-based features for both dysarthria detection and assessment. They showed speakers with dysarthria tend to have irregular rhythmic patterns that turn out to be useful for detecting dysarthria.

However, none of the previous studies considered the mismatch between training and testing data. To the best of our knowledge, this is the first work in which domain adaptation is applied to dysarthria detection. We take into account the case in which we have access to J dysarthric datatsets with different types of noise. While the domain adaptation problem for dysarthric speech has not been addressed yet, the adaptation of a model to a new unseen speech dataset with a different noise type is an old problem that always attracted the attention of the ASR community [145–147].

Due to the fact that both dysarthria detection and noise robustness are challenging tasks, they tend to be treated separately. However, it is worthwhile to consider them as a unique problem. Indeed, dysarthric speech datasets are usually collected at the hospitals that are not equipped to record speech in absence of noise. Moreover, due to the difficulties of collecting dysarthric speech only few corpora are available and, due to their limited size, they often cannot be employed separately to train a dysarthria detector. We attempt to overcome this problem by leveraging many dysarthric datasets at the same time in order to a learn a binary classifier for an unseen and unlabelled dataset.

Dataset TORGO is one of the most popular dysarthric speech datasets [14]. It consists of aligned acoustic and articulatory recordings from 15 speakers. Seven of these speakers are control speakers (4 males, 3 females) without any speech disorders. The remaining eight speakers (5 males, 3 females) present different levels of dysarthria, diagnosed by speech-language pathologists based on the Frenchay dysarthria assessment [1]. The subjects were asked to read single words or sentences and to describe the content of some photos. This procedure was repeated many times for each speaker resulting in approximately three hours of speech. We do not consider unrestricted words and we also discard about some corrupted files. Table 3.1 summarizes the database characteristics. Note that the number of utterances and .wav files do not correspond. This is because for some utterance the recordings are done by using multiple microphones (arrayMic and headMic).

TORGO	**Dysarthric speakers**								**Control speakers**						
	F01	**M01**	**M02**	**M04**	**M05**	**F03**	**F04**	**M03**	**FC01**	**FC02**	**FC03**	**MC01**	**MC02**	**MC03**	**MC04**
Disorder	Severe	Severe	Severe	Severe	S-M	Moderate	Mild	Mild	None	None	None	None	None	None	None
N. utterances	118	372	389	395	480	559	434	410	152	1228	974	1078	677	872	999
N. wav files	236	743	774	673	592	1107	681	816	304	2194	1938	2155	1123	1673	1624

Table 3.1: F: female speaker, M: male speaker, FC: female control speaker, MC: male control speaker; S-M represents severe-moderate category of dysarthria patients

We used the TORGO dataset to generate multiple-sources and target domain. In particular, we generated 15 noisy datasets by combining the raw tracks with different types of noises from a noise dataset[1]. The noisy datasets have been synthesized by PyDub python library [148]. We then used the libROSA Python library [149] to extract 13 MFCCs plus deltas and delta-deltas, computed every 10ms from 25ms Hamming windows followed by a z-normalization per track.
We fixed the number of sources equal to 14 and we tested 4 noisy domains as target: F16, Buccaneer2 (B2), Factory2 (F2), Destroyerengine (D).

[1] Available at `http://spib.linse.ufsc.br/noise.html`

Compared methods We compare the MSDA methods with the `Baseline` case in which a unique classifier is trained on all the sources and tested on the target domain, without any adaptation procedure (for more details, see Sec. 3.4.1, Eq. 3.13).
We then report the performances of the two JDOT extensions previously described in Sec 3.4.2, that are MJDOT and CJDOT. For these and for MSDA-WJDOT, we investigate both SSE and ACC validation strategies. If not specified, we consider the validation through the SSE (`MSDA-WJDOT/CJDOT/MDJOT`), otherwise we add a superscript (`MSDA-WJDOT`^{ACC}/ `CJDOT`^{ACC}/ `MDJOT`^{ACC}).
Also, we recall that in `MSDA-WJDOT` the embedding function g is provided by the `Baseline`, while in `MSDA-WJDOT`$_{MTL}$ we employ the MTL approach of formula 3.12 (and similarly for MJDOT and CJDOT).
In addition to the JDOT extensions, we compare the proposed method with Importance Weighted Empirical Risk Minimization (`IWERM`) [132] already introduced in Sec. 3.4.2. The aforementioned well-established MSDA methods do not require the learning of an embedding and, thus, they provide a meaningful comparison with the proposed approach. However, the following step of this work will be to investigate comparisons with a broader range of state-of-the-art MSDA models, such as DCTN [100], DARN [99], MDAN [98].

MSDA-WJDOT model details The feature extraction g is performed by a Bidirectional Long Short-Term Memory (BLSTM) recurrent network with two hidden layers, each containing 50 memory blocks. We train the BLSTM as sequence-to-vector model. The source and target classifier functions $\{f_j\}$,f_S, f are learned as one feed-forward layer taking the last hidden state as input.

The proposed method MSDA-WJDOT is a two-step procedure. Firstly, we learn the embedding function g. In the `MSDA-WJDOT`$_{MTL}$, it is learned simultaneously with the source-classifiers $\{f_j\}$ following Eq. 3.12. Instead, in `MSDA-WJDOT` we learn g with a global source classifier f_S. Then, g is kept fixed and the target classifier f is learned in an unsupervised way by minimizing the Wasserstein distance as in Eq. 3.10. The adopted model is shown in Fig. 3.12. The weights were initialized with Xavier initialization. Training is performed with Adam optimizer with 0.9 momentum and $\epsilon = e^{-8}$. The learning rate exponentially decays at every epoch. We grid-research the initial learning rate value and the decay rate.

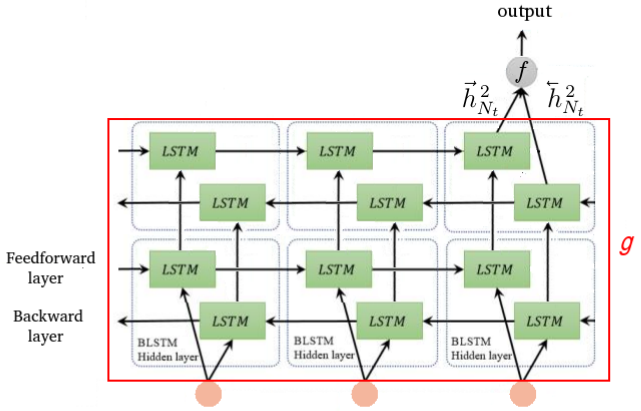


Figure 3.12: The g function is provided by the BLSTM architecture in the red rectangle. During the second phase, g is kept fixed and MSDA-WJDOT learns the target classification f.

Results In all experiments, we observed that learning g by the Multi-Task Learning approach always provides a better performance in both JDOT extensions and MSDA-WJDOT. Hence, for an easier reading of Table 3.2, we only report the performances in which the extractor g is given by the MTL.

For all the target domains, `IWERM` completely fails underperforming also the `Baseline`. This is probably due to the difficulties in computing the probability density function of the acoustic input, that presents a high complexity. Indeed, to make the computation feasible, we firstly extracted a lower-dimensional vector from the audio signal by PCA.

All the remaining state-of-the-art DA methods outperform the Baseline but MSDA-WJDOT provides the best accuracy for all target domains, with both validation strategies. More precisely, MSDA-WJDOT provides a relative improvement of 4.9% and 0.9% over the accuracy of the `Baseline` and the best competitor model, respectively.

Target domain	F16	B2	F2	D	Average
`Baseline`	93.59 ± 0.38	93.76 ± 0.22	93.23 ± 0.66	92.46 ± 0.82	93.26
`IWERM` [132]	66.22 ± 0.01	66.38 ± 0.01	66.25 ± 0.05	66.30 ± 0.09	66.29
$\texttt{CJDOT}_{MTL}$ [85]	96.49 ± 0.09	97.37 ± 0.08	96.53 ± 0.08	94.36 ± 0.09	96.19
$\texttt{CJDOT}^{Acc}_{MTL}$ [85]	95.35 ± 0.55	97.39 ± 0.09	96.71 ± 0.07	92.98 ± 0.75	95.61
$\texttt{MJDOT}_{MTL}$ [85]	96.42 ± 0.14	97.29 ± 0.05	96.49 ± 0.14	94.30 ± 0.07	96.13
$\texttt{MJDOT}^{Acc}_{MTL}$ [85]	95.81 ± 0.42	97.22 ± 0.09	96.53 ± 0.12	93.12 ± 0.67	95.67
$\texttt{MSDA-WJDOT}_{MTL}$	96.54 ± 0.17	97.61 ± 0.06	96.51 ± 0.17	94.80 ± 0.13	96.37
$\texttt{MSDA-WJDOT}^{Acc}_{MTL}$	**97.32 ± 0.36**	**97.82 ± 0.13**	**97.76± 0.10**	**95.42 ± 0.24**	**97.08**

Table 3.2: Dysarthria detection accuracy on four target datasets: F16, Buccaneer2 (B2), Factory2 (F2), Destroyerengine (D). The mean and standard deviation of the accuracy are reported for the Baseline, the two extensions of JDOT and the proposed MSDA-WJDOT approach with two early stopping strategies (SSE and ACC).

3.5.2 Unsupervised Dysarthric Speaker Adaptation in Spoken Command Recognition

In the following, we address the problem of speaker adaptation in ASR systems. In particular, we consider a Spoken Command Recognition that is, roughly speaking, a small-vocabulary ASR model. We apply the proposed MSDA-WJDOT algorithm to perform unsupervised speaker adaptation when multiple labelled speaker datasets are available. We focus on the case in which the target speaker is dysarthric, while source speakers can be both healthy and dysarthric individuals.

Previous works Even though speaker-adaptation (SA) techniques have been widely investigated by the speech community, conventional SA algorithms perform poorly in dysarthric speech when they present a low intelligibility. As reported in [150–152], even though commercial ASR systems can achieve up to 90% for some dysarthric speakers with high intelligibility, the recognition performance still remains inadequate with the decreasing of the speech intelligibility. These commercial systems usually incorporate SA techniques to adapt the model to the voice of that speaker that require some audio samples from the speaker of interest. It follows that traditional adaptation techniques are not sufficient to deal with gross abnormalities [153]. Further, they results to be inefficient even for speakers with mild to moderate dysarthria when the vocabulary size is larger than 30 words [154].

In the last decades, some attempts to move forward and solve the training-testing mismatch for pathological speech have been done. [155] adapts the ASR system trained on large datasets to a dysarthric dataset. In [60], the authors leverage articulatory features, as well as the acoustic ones, achieving a 4-8% of World Error Rate (WER) relative reduction. However, these techniques depend on the amount of data available for fine-tuning, that is usually limited. To overcome this problem, [156] proposes to fine-tune only a subset of the network layers to better adapt an ASR model to ALS speech. [157], rather than adapting the acoustic models, models the errors at phonetic level made by the speaker and attempts to correct them by two possible strategies, that incorporate the language model and find the best estimate of the correct word sequence.

However, all the aforementioned studies are limited to the supervised framework, whereas the unsupervised speaker adaptation is the most common real scenario.

Dataset To investigate the Dysarthric Speaker Adaptation we employ the AllSpeak dataset [42], that will be discuss in depth in the next chapter. It consists of speech recordings from 29 Italian native speakers. Seventeen of these are affected by Amyotrophic Lateral Sclerosis, while the remaining thirteen are speakers of control. The dataset contains 25 commands in italian, relative to basic needs such as “I am thirsty”.

This dataset is very challenging due to the small amount of recordings. Indeed, only 2387 and 1857 examples have been recorded from control and dysarthric speakers, respectively. The number of recorded commands for each speaker is shown in Table 3.3.

Control speaker	MCO1	FC01	FC02	MC02	MC03	FC03	MC04	FC04	MC05	FC05	MC06	FC06
N. commands	142	175	245	300	249	226	129	146	223	161	230	161

Dysarthric speaker	M01	M02	F01	F02	M03	M04	M05	F03	M06	M07	M08	M09	F04	M10	M11	M12	M13
N. commands	119	117	57	125	45	125	126	115	115	125	125	119	114	125	125	125	55

Table 3.3: AllSpeak Dataset

MSDA-WJDOT model details The feature extraction g is performed by a Bidirectional Long Short-Term Memory (BLSTM) recurrent network with five hidden layers, each containing 250 memory blocks. Finally, a softmax layer performs the classification task. Here, we do not consider the MTL variant as the dataset size is limited and learning a source-classifier f_j with a very small amount of data may result hard.

All weights were initialized with Xavier initialization. Training is performed with Adam optimizer with 0.9 momentum and $\epsilon = e^{-8}$. Learning rate is fixed to 0.001. We perform speaker adaptation of each dysarthric speaker by using all the remaining speakers as training dataset. Within this, a dysarthric subset (one example of each command for each speaker) is used as validation set to perform Early stopping.

Results We compare the proposed MSDA method with the `Baseline`, described in Sec. 3.4.1 (Eq. 3.13), in which a classifier is trained on all sources and directly tested, without any adaptation, on the target domain. Further, we compare the Speaker Adaptation performance with the two JDOT extensions, `MJDOT` and `CJDOT`.

Table 3.4 reports the results in terms of Command Error Rate (CER). A first remark is that, even though the `Baseline` always achieved a CER between 15% and 20% on the validation set, it often has low performances on the target speaker. Once again, this emphasizes the difficulty of an ASR system to recognize the speech of a new dysarthric speaker and the importance of the speaker adaptation in this context.
The unsupervised speaker adaptation carried out by `MSDA-WJDOT` outperforms all the methods by providing the best Average CER. Indeed, it reduces the CER of 21% over the `Baseline`. Further, we provide the Average Rank that is a performance measure suitable when several targets domain are tested. To every method, it assigns a rank (from 1 to 4, that is the number of considered methods) for each tested target based on the CER (e.g., 1 if the method has the lowest CER, 4 for the highest CER) and then it computes the average of the ranks. This measure is more robust to the variance and confirms that `MSDA-WJDOT` provides the best performance.

It is crucial to recall that `MSDA-WJDOT` also equip us with a measure of similarity between the target and the sources and, hence, between speakers. We found that the recovered $\boldsymbol{\alpha}$ always attributes highest similarity scores to dysarthric speakers rather than healthy ones, suggesting that this approach can realistically estimate speaker closeness. Fig. 3.13 and 3.14 show the recovered weights for target speaker M05 and M13, respectively. As we can see, in both cases, the mass is spread along dysarthric speakers while the α_j values are close to zero for healthy speakers. To further support this hypothesis, we show in Fig. 3.15 the weights recovered by MSDA-WJDOT when the target speaker is a control speaker. In this case, the algorithm points out control speakers as the most similar ones.

Target Speaker	Baseline	CJDOT [85]	MJDOT [85]	MSDA-WJDOT
M01	35.79	32.77	**31.93**	**31.93**
M02	**34.26**	36.75	36.75	37.71
F01	63.16	52.63	**49.12**	**49.12**
F02	48.50	**40.00**	**40.00**	**40.00**
M03	64.44	68.89	71.11	**57.78**
M04	**30.00**	31.20	32.00	30.40
M05	18.62	17.46	15.08	**14.29**
F03	68.33	**61.74**	70.43	62.61
M06	48.67	34.78	**33.91**	35.65
M07	11.00	**7.20**	8.00	8.80
M08	39.50	36.00	41.60	**33.60**
M09	24.79	**16.81**	18.49	19.33
F04	48.07	**38.60**	**38.60**	**38.60**
M10	18.00	12.80	**12.00**	12.80
M11	56.50	47.20	48.80	**45.60**
M12	7.50	5.60	7.20	**4.80**
M13	30.91	45.45	43.64	**21.82**
Average CER	38.11	34.46	34.37	**30.09**
Average Rank	2.88	1.94	2.29	**1.65**

Table 3.4: Command Error Rate (CER) for each dysarthric target speaker provided by the Baseline, MSDA-WJDOT and the competitor models.

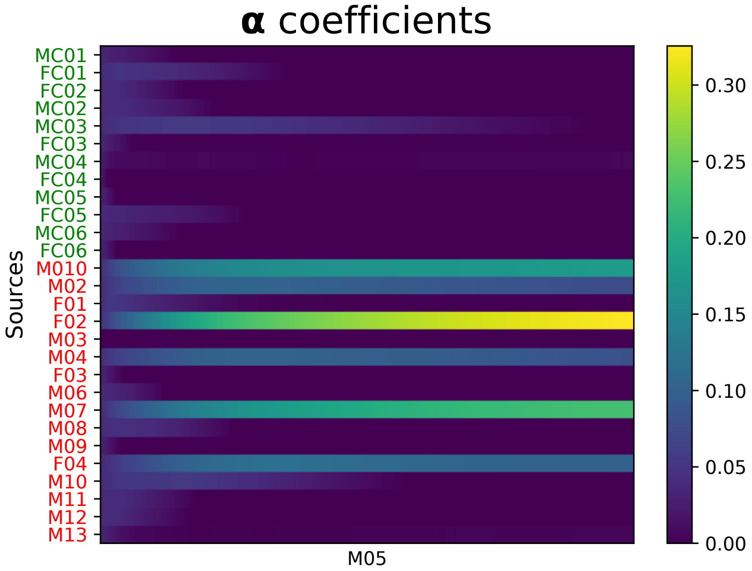


Figure 3.13: $\boldsymbol{\alpha}$ coefficients recovered during MSDA-WJDOT training for target speaker M05. The α_j coefficients are close to zero for healthy speakers (in green), while the highest weights are attributed to dysarthric speakers (in red).

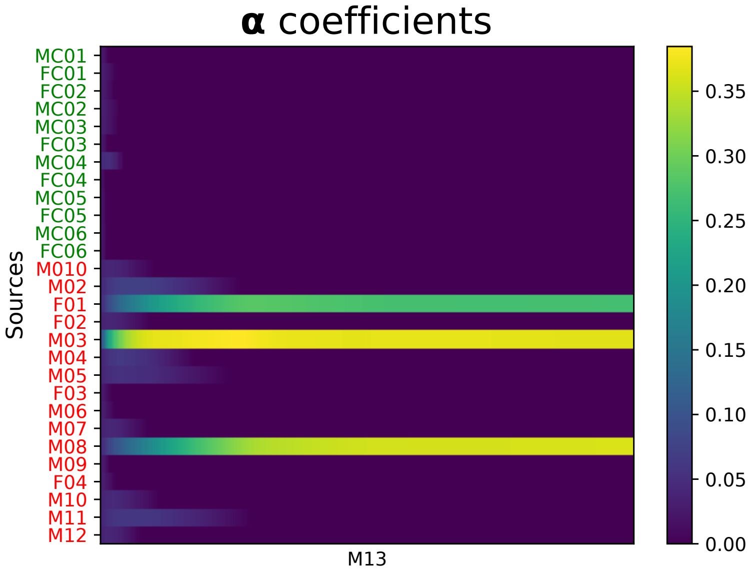


Figure 3.14: $\boldsymbol{\alpha}$ coefficients recovered during MSDA-WJDOT training for target speaker M13. The α_j coefficients are close to zero for healthy speakers (in green), while the highest weights are attributed to dysarthric speakers (in red)

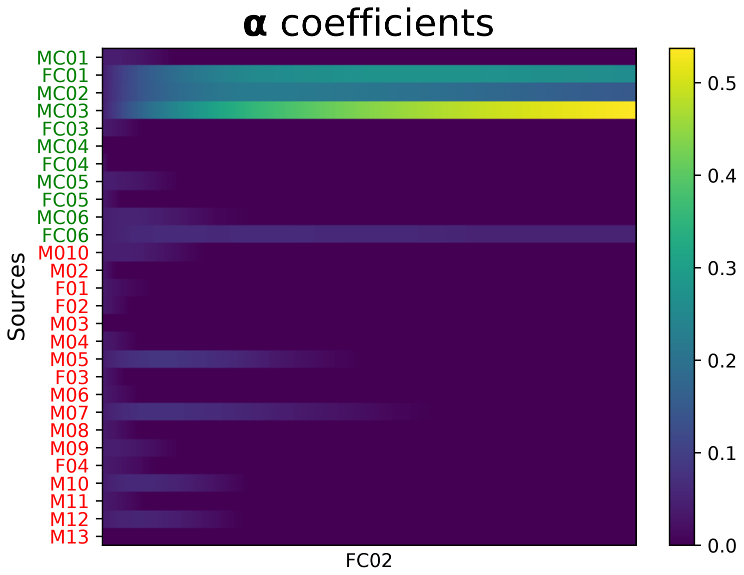


Figure 3.15: $\boldsymbol{\alpha}$ coefficients recovered during MSDA-WJDOT training when the target speaker is a control speaker. Contrary to the previous cases, the α_j coefficients are close to zero for dysarthric speakers (in red), while the highest weights are attributed to controls speakers (in red)

The $\boldsymbol{\alpha}$ weight and the dysarthria detection As we showed in Fig. 3.13, 3.14, 3.15, the MSDA-WJDOT algorithm associates speakers with similar voice characteristics. We took advantage of this property to move a step forward, that is leveraging the $\boldsymbol{\alpha}$ weight to detect dysarthria. Specifically, we attempt to classify a speaker as healthy or dysarthric based on her similarity with the other subjects.

Let define I_c as the set indexing the control speakers and I_d as the set of indices related to dysarthric speakers. We then define the Healthy Score (HS) and the Dysarthric Score (DS) as follow

$$HS = \sum_{j \in I_c} \alpha_j \tag{3.17}$$

$$DS = \sum_{j \in I_d} \alpha_j. \tag{3.18}$$

We can use these scores to perform dysarthria detection by stating that

A speaker is affected by dysarthria if

$$DS > HS.$$

Fig. 3.16 reports the computed scores for all dysarthric speakers and for 5 control speakers. As we can see the controls subjects are always classified as healthy while for the patients, except for F02, we have $DS > HS$. This results in a final accuracy of

95%. In 3.17, we report the confusion matrix, where H. and D. stands for Healthy and Dysarthric, respectively.

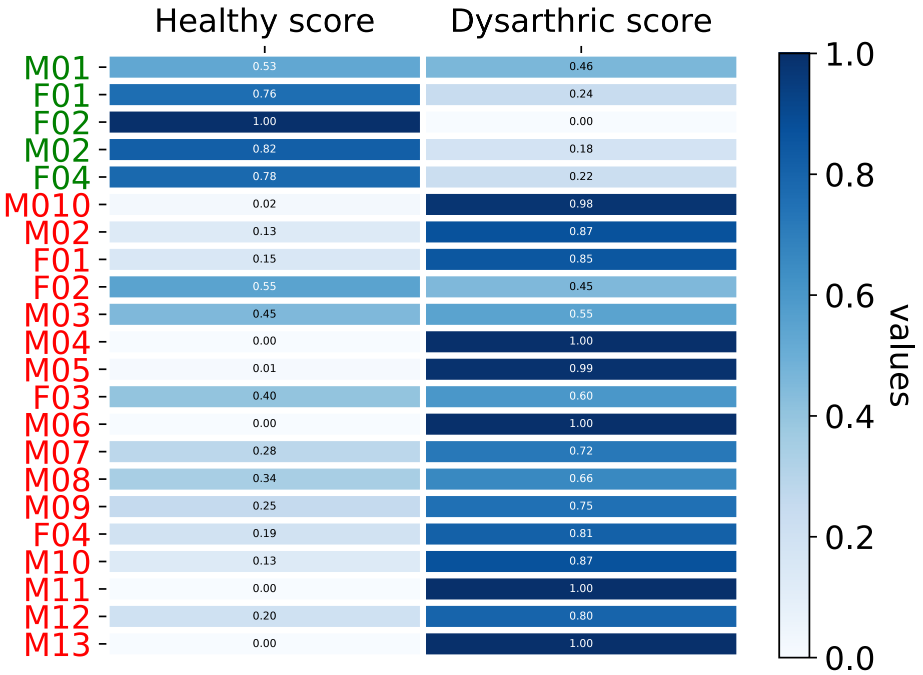


Figure 3.16: HS and DS computed for healthy (in green) and dysarthric (in red) speakers.

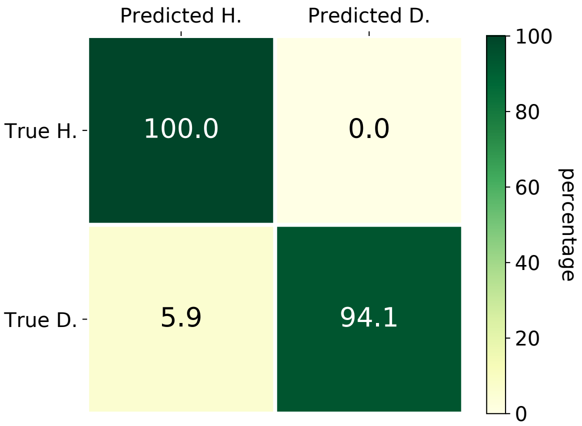


Figure 3.17: Confusion matrix.

3.6 Discussion

In this work, we developed an algorithm to confront multi-sources domain adaptation. We supported it with theoretical guarantees and we showed its effectiveness on both simulated and real data. More precisely, we employed it on domain adaptation for dysarthria detection and dysarthric speaker adaptation for ASR. For the thesis purpose we only reported the applications on dysarthric speech but it may be adopted in other contexts, such as object recognition, music-speech discrimination, sentiment analysis, etc.

The big advantage of this approach w.r.t. the other state-of-the-art methods is that it provides a closeness measures between sources and target distributions, as well as a classifier for the target domain. The similarity measure, provided by the $\boldsymbol{\alpha}$ weight, allows us to select only the relevant sources by discarding useless or misleading data. Future studies could leverage this weight to reduce the time consumption by using only the selected sources during the training algorithm.

Also, in this thesis, we did not delve into the learning of the embedding. The first step of future work will be the comparison with state-of-the-art methods in which the embedding is simultaneously learned with the classifier [98–101]. Further, we will investigate more in depth the role of the embedding in our approach, as results showed that $\texttt{MSDA-WJDOT}_{MTL}$ always outperforms `MSDA-WJDOT`. In Sec. 3.5.2, for instance, we did not adopt the MTL approach for the embedding extraction due to the limited size of the dataset. However, one may think some strategies to overcome this issue such as the pre-training of the network on a large dataset or data augmentation.

Moreover, we here focused on unsupervised adaptation but MSDA-WJDOT can be easily extended to semi-supervised learning in which the embedding g and the sources classifiers are learned simultaneously with $\boldsymbol{\alpha}$ and the target classifier.

Results reported in Sec. 3.5 stress out the benefits of the proposed method in a challenging context, that is the one involving pathological speech. Indeed, the proposed algorithm turned out to be effective when sources and target are dysarthric corpora with different noises, outperforming both the Baseline and the other state-of-the-art MSDA models. Further, we employed the MSDA-WJDOT algorithm to confront the dysarthric speaker adaptation problem. It provided the best performance and a remarkable Command Error Rate (CER) reduction of 21% over the Baseline. Most interestingly, such an approach is based on the estimation of source-target similarity coefficients $\boldsymbol{\alpha}$ that are a measure of speaker relatedness.
We observed high $\boldsymbol{\alpha}$ values for homogeneous source/target speakers (i.e., both healthy or both dysarthric), whereas we have smaller weights for mismatched source/target speakers (e.g., healthy speaker in the source domain and dysarthric speaker in the target one). Thus, the closeness in the Wasserstein sense reflects the closeness in the speech characteristics. We took advantage of this relationship to define a Healthy and a Dysarthric Index. The first one is computed by summing all the $\boldsymbol{\alpha}$ weights attributed to healthy source speakers, the second one by summing the coefficients related to dysarthric source speakers. We used these indices to predict if the target speaker was affected by dysarthria by looking at the highest index. This simple procedure allowed us to detect dysarthria with the 95% of accuracy.

Future directions could delve into the study of the Dysarthric Index (DI) and its correlation with the assessed dysarthria severity. In this work, we diagnosed the

dysarthria for a DI higher than 0.5. However, this constraint may be too rigid and, for instance, a DI value between 0 and 0.5 may indicate a mild dysarthria.
Moreover, it would be interesting to find a correspondence between intervals of DI and dysarthria severity levels. Looking to the future, this may bring us to very efficient ASR systems that simultaneously improve their performance via the speaker-adaptation, and compute the DI warning the subject when the index is close to the interval associated to the next severity level. Hence, such a device could predict the disease degeneration and allow the patient to act in time in order to prevent it.

3.7 Appendix: Proofs

We provide proof of Lemma 1 and Theorem 1 introduced in Sec. 3.3.1. For reader's convenience the results are here repeated.

3.7.1 Proof of Lemma 1

Lemma 1. *For an hypothesis $f \in \mathcal{H}$, denote as $\varepsilon_{p_T}(f)$ and $\varepsilon_{p_S^{\boldsymbol{\alpha}}}(f)$, the expected loss of f on the target and on the weighted sum of the source domains, with respect to a loss function L bounded by B. We have*

$$\varepsilon_{p_T}(f) \leq \varepsilon_{p_S^{\boldsymbol{\alpha}}}(f) + B \cdot D_{TV}\left(p_S^{\boldsymbol{\alpha}}, p_T\right) \tag{3.19}$$

where $p_S^{\boldsymbol{\alpha}} = \sum_{j=1}^{J} \alpha_j p_{S,j}$ with $\boldsymbol{\alpha} \in \Delta^J$ is a convex combination of the source distributions, and D_{TV} is the total variation distance.

Proof. We define the error of an hypothesis f with respect to a loss function $L(\cdot,\cdot)$ and a joint probability distribution $p(x,y)$ as

$$\varepsilon_p(f) = \int p(x,y)L(y,f(x))dxdy$$

then using simple arguments, we have

$$\begin{aligned}
\varepsilon_{p_T}(f) &= \varepsilon_{p_T}(f) + \varepsilon_{p_S^{\boldsymbol{\alpha}}}(f) - \varepsilon_{p_S^{\boldsymbol{\alpha}}}(f) \\
&\leq \varepsilon_{p_S^{\boldsymbol{\alpha}}}(f) + |\varepsilon_{p_T}(f) - \varepsilon_{p_S^{\boldsymbol{\alpha}}}(f)| \\
&\leq \varepsilon_{p_S^{\boldsymbol{\alpha}}}(f) + \int |p_S^{\boldsymbol{\alpha}}(x,y) - p_T(x,y)||L(y,f(x)|dxdy \\
&\leq \varepsilon_{p_S^{\boldsymbol{\alpha}}}(f) + B \int \left|p_S^{\boldsymbol{\alpha}}(x,y) - p_T(x,y)\right|dxdy
\end{aligned} \tag{3.20}$$

and using the definition of the total variation distance between distribution we conclude the proof. □

3.7.2 Proof of Theorem 1

The proof of this theorem follows the same steps as the one proposed by Courty et al. [85] and we reproduce it here for a sake of completeness.

Definition 2. (Probabilistic Transfer Lipschitzness) *Let p_S and p_T be respectively the source and target distributions. Let $\phi : \mathbb{R} \to [0,1]$. A labeling function $f : \mathcal{G} \to \mathbb{R}$ and a joint distribution $\pi \in \Pi(p_S, p_T)$ over p_S and p_T are Φ-Lipschitz transferable if for all $\lambda > 0$, we have*

$$\mathrm{Prob}_{(x_S,x_T)\sim\pi}\big[|f(x_S) - f(x_T)|| > \lambda D(x_S, x_T)\big] \leq \Phi(\lambda)$$

with D being a metric on $\mathcal{G}$

As stated in Courty et al. [85], given function f and coupling π, this property and definition gives a bound on the probability of finding couple (source-target) of examples that are differently labelled in a $(1/\lambda)$-ball with respect to π and the metric D .

Lemma 2. *Let $\mathcal{H}$ be a space of M-Lipschitz labelling functions. Assume that, for every $f \in \mathcal{H}$ and $x \in \mathcal{G}$, $|f(x) - f(x')| \leq M$. Consider the following measure of similarity between p_S^{α} and p_T introduced in [127, Def. 5]*

$$\Lambda(p_S^{\alpha}, p_T) = \min_{f\in\mathcal{H}} \varepsilon_{p_S^{\alpha}}(f) + \varepsilon_{p_T}(f), \tag{3.21}$$

where the risk is measure w.r.t. to a symmetric and k-Lipschitz loss function that satisfies the triangle inequality. Further, assume that the minimizing function f^* satisfies the Probabilistic Transfer Lipschitzness (PTL) property. Then, for any $f \in \mathcal{H}$, we have

$$\varepsilon_{p_T}(f) \leq W_D\left(p_S^{\alpha}, p_T^{f}\right) + \Lambda(p_S^{\alpha}, p_T) + kM\phi(\lambda), \tag{3.22}$$

where $\phi(\lambda)$ is a constant depending on the PTL of $f^{\star}$.

Proof. We have that

$$\begin{aligned}
\varepsilon_{p_T}(f) &:= \mathbb{E}_{(x,y)\sim p_T} L(y, f(x)) \\
&\leq \mathbb{E}_{(x,y)\sim p_T}\big[L(y, f^{\star}(x)) + L(f^{\star}(x), f(x))\big] \\
&= \varepsilon_{p_T}(f^{\star}) + \mathbb{E}_{(x,y)\sim p_T} L(f^{\star}(x), f(x)) \\
&= \varepsilon_{p_T}(f^{\star}) + \mathbb{E}_{(x,y)\sim p_T^{f}} L(f^{\star}(x), f(x)) \\
&= \varepsilon_{p_T}(f^{\star}) + \varepsilon_{p_T^{f}}(f^{\star}) + \varepsilon_{p_S^{\alpha}}(f^{\star}) - \varepsilon_{p_S^{\alpha}}(f^{\star}) \\
&\leq |\varepsilon_{p_T^{f}}(f^{\star}) - \varepsilon_{p_S^{\alpha}}(f^{\star})| + \varepsilon_{p_S^{\alpha}}(f^{\star}) + \varepsilon_{p_T}(f^{\star})
\end{aligned}$$

where the second equality comes from the symmetry of the loss function and the third one is due to the fact that $\mathbb{E}_{(x,y)\sim p_T} L(f^{\star}(x), f(x)) = \mathbb{E}_{(x,y)\sim p_T^{f}} L(f^{\star}(x), f(x)) = \mathbb{E}_{x\sim\mu_T} L(f^{\star}(x), f(x))$ since the label y is not used in the expectation.

Now, we analyze the first term, note we have that samples drawn from p_T^f distribution can be expressed as $(x_T, y_T^f) \sim p_T^f$ with $y_T^f = f(x_T)$.

$$
\begin{aligned}
&|\varepsilon_{p_T^f}(f^\star) - \varepsilon_{p_S^\alpha}(f^\star)| \\
=&\left| \int_{\mathcal{G}\times\mathbb{R}} L(y, f^\star(x))(p_T^f(x,y) - p_S^\alpha(x,y))dxdy \right| \\
=&\left| \int_{\mathcal{G}\times\mathbb{R}} L(y, f^\star(x))d(p_T^f - p_S^\alpha) \right| \\
\leq& \int_{(\mathcal{G}\times\mathbb{R})^2} |L(y_T^f, f^\star(x_T)) - L(y_\alpha, f^\star(x_\alpha))| d\pi^\star((x_\alpha, y_\alpha), (x_T, y_T^f)) \qquad (3.23) \\
=& \int_{(\mathcal{G}\times\mathbb{R})^2} \Big[L(y_T^f, f^\star(x_T)) - L(y_T^f, f^\star(x_\alpha)) \\
&\qquad + L(y_T^f, f^\star(x_\alpha)) - L(y_\alpha, f^\star(x_\alpha)) \Big] d\pi^\star((x_\alpha, y_\alpha), (x_T, y_T^f)) \\
\leq& \int_{(\mathcal{G}\times\mathbb{R})^2} \Bigg[\Big| L(y_T^f, f^\star(x_T)) - L(y_T^f, f^\star(x_\alpha)) \Big| \\
&\qquad + \Big| L(y_T^f, f^\star(x_\alpha)) - L(y_\alpha, f^\star(x_\alpha)) \Big| \Bigg] d\pi^\star((x_\alpha, y_\alpha), (x_T, y_T^f)) \\
\leq& \int_{(\mathcal{G}\times\mathbb{R})^2} \Bigg[k|f^\star(x_T) - f^\star(x_\alpha))| \\
&\qquad + \Big| L(y_T^f, f^\star(x_\alpha)) - L(y_\alpha, f^\star(x_\alpha) \Big| \Bigg] d\pi((x_\alpha, y_\alpha), (x_T, y_T^f)) \qquad (3.24) \\
\leq& kM\phi(\lambda) + \int_{(\mathcal{G}\times\mathbb{R})^2} \Bigg[k\lambda D(x_T, x_\alpha) \\
&\qquad + \Big| L(y_T^f, f^\star(x_\alpha)) - L(y_\alpha, f^\star(x_\alpha)) \Big| \Bigg] d\pi((x_\alpha, y_\alpha), (x_T, y_T^f)) \qquad (3.25) \\
\leq& kM\phi(\lambda) + \int_{(\mathcal{G}\times\mathbb{R})^2} \Bigg[\beta D(x_T, x_\alpha) + \Big| L(y_T^f, y_\alpha) \Big| \Bigg] d\pi((x_\alpha, y_\alpha), (x_T, y_T^f)) \\
& \qquad (3.26) \\
=& kM\phi(\lambda) + W_1(p_S^\alpha, p_T^f). \qquad (3.27)
\end{aligned}
$$

Inequality in line (3.23) is due to the Kantorovich-Rubinstein theorem stating that for any coupling $\pi \in \Pi(p_S^\alpha, p_T)$ the following inequality holds

$$
\begin{aligned}
&\left| \int_{\mathcal{G}\times\mathbb{R}} L(y, f^\star(x))d(p_T^f - p_S^\alpha) \right| \\
\leq& \left| \int_{(\mathcal{G}\times\mathbb{R})^2} |L(y_T^f, f^\star(x_T)) - L(y_\alpha, f^\star(x_\alpha)| d\pi((x_\alpha, y_\alpha), (x_T, y_T^f)) \right|,
\end{aligned}
$$

followed by an application of the triangle inequality. Since, the above inequality applies for any coupling, it applies also for $\pi^\star$. Inequality (3.24) is due to the assumption that the loss function is k-Lipschitz in its second argument. Inequality (3.25) derives from the PTL property with probability $1-\phi(\lambda)$ of $f^\star$ and $\pi^\star$. In addition, taking into account that the difference between two samples with respect to $f^\star$ is bounded by M, we have the term $kM\phi(\lambda)$ that covers the regions where PTL assumption does not hold. Inequality (3.26) is obtained from the symmetry of $D(\cdot,\cdot)$, the triangle inequality on the loss and by posing $k\lambda = \beta$. □

First we need to prove the following Lemma.

Lemma 3. *For any distributions $\hat{p}_{S,j}, p_{S,j}$ and $\boldsymbol{\alpha} \in \Delta^J$ in the simplex we have*

$$W_D\left(\sum_j \alpha_j \hat{p}_{S,j}, \sum_j \alpha_j p_{S,j}\right) \leq \sum_j \alpha_j W_D\left(\hat{p}_{S,j}, p_{S,j}\right).$$

Proof. First we recall that the Wasserstein Distance between two distribution is

$$W_D(p,p') = \min_{\pi\in\Pi(p,p')} \int D(\mathbf{v},\mathbf{v}')\pi(\mathbf{v},\mathbf{v}')d\mathbf{v}d\mathbf{v}', \tag{3.28}$$

where $\Pi(p,p') = \{\pi | \int \pi(\mathbf{v},\mathbf{v}')d\mathbf{v}' = p(\mathbf{v}), \int \pi(\mathbf{v},\mathbf{v}')d\mathbf{v} = p'(\mathbf{v}')\}$. Let $\pi^*_{S,j}$ be the optimal OT matrix between $\hat{p}_{S,j}$ and $p_{S,j}$. It is obvious to see that $\sum_j \alpha_j \pi^*_{S,j}$ respects the marginal constraints for $W_D\left(\sum_j \alpha_j \hat{p}_{S,j}, \sum_j \alpha_j p_{S,j}\right)$, i.e. $\sum_j \alpha_j \pi^*_{S,j} \in \Pi\left(\sum_j \alpha_j \hat{p}_{S,j}, \sum_j \alpha_j p_{S,j}\right)$. Hence, $\sum_j \alpha_j \pi^*_{S,j}$ is a feasible solution for the OT problem and, consequently, the cost for this feasible solution is greater or equal than the optimal value $W_D\left(\sum_j \alpha_j \hat{p}_{S,j}, \sum_j \alpha_j p_{S,j}\right)$. Since $\int D(\mathbf{v},\mathbf{v}') \sum_j \alpha_j \pi^*_{S,j}(\mathbf{v},\mathbf{v}')d\mathbf{v}d\mathbf{v}' = \sum_j \alpha_j W_D\left(\hat{p}_{S,j}, p_{S,j}\right)$ we recover the Lemma above. □

We can now prove Theorem 1, which we also restate for the convenience of the reader.

Theorem 1. *Under the assumptions of Lemma 2, let $\hat{p}_{S,j}$ be j-th source empirical distributions of N_j samples and $\hat{p}_T$ the empirical target distribution with N_T samples. Then for all $\lambda > 0$, with $\beta = \lambda k$ in the ground metric D we have with probability $1-\eta$*

$$\varepsilon_{p_T}(f) \leq W_D\left(\hat{p}^{\boldsymbol{\alpha}}_S, \hat{p}^f_T\right) + \sqrt{\frac{2}{c'}\log\frac{2}{\eta}}\left(\frac{1}{N_T} + \sum_j \frac{\alpha_j}{N_j}\right) + \Lambda(p^{\boldsymbol{\alpha}}_S, p_T) + kM\phi(\lambda). \tag{3.29}$$

Proof. In order to prove Theorem 2 first we show that

$$
\begin{aligned}
W_D\left(\sum_j \alpha_j p_{S,j}, p_T^f\right) &\leq W_D\left(\sum_j \alpha_j \hat{p}_{S,j}, \hat{p}_T^f\right) \\
&\qquad + W_D(\hat{p}_T^f, p_T^f) + W_D\left(\sum_j \alpha_j \hat{p}_{S,j}, \sum_j \alpha_j p_{S,j}\right) \\
&\leq W_D\left(\sum_j \alpha_j \hat{p}_j, \hat{p}_T^f\right) + W_D(\hat{p}_T^f, p_T^f) + \sum_j \alpha_j W_D\left(\hat{p}_{S,j}, p_{S,j}\right)
\end{aligned}
$$

where the last line is obtained from Lemma 3. Using the well known convergence property of the Wasserstein distance proven in [158] we find the following bound with probability $1-\eta$

$$
\varepsilon_{p_T}(f) \leq W_D\left(\sum_j \alpha_j \hat{p}_{S,j}, \hat{p}_T^f\right) + \sqrt{\frac{2}{c'}\log\left(\frac{2}{\eta}\right)}\left(\frac{1}{N_T} + \sum_j \frac{\alpha_j}{N_j}\right) + \Lambda(p_S^{\boldsymbol{\alpha}}, p_T) + 2kM\phi(\lambda) \tag{3.30}
$$

with c' corresponding to all source and target distributions under similar conditions as in [85]. □

Chapter 4

Small vocabulary ASR for dysarthric speech

In this Chapter, we focus on small-vocabulary ASR systems and, in particular, on Voice Command Recognition. Specifically, we first describe a database of command speech that we recorded from 20 healthy and 21 dysarthric individuals. The recordings consist of mobile phone commands to make calls and menage a Contact smartphone application, such as “start Contacts”, “call”, “end call”. Then, we exploit an Italian command dataset recorded from people affected by Amyotrophic Lateral Sclerosis (ALS) to develop a Voice Command Recognition. This model has been integrated in an Android Application that helps ALS patients to communicate their basic needs, especially when their speech intelligibility is almost vanished.

These projects, i.e. the collection of a speech command corpus and the development of a Voice Command recognition for dysarthric speech, aim at providing two different tools. In one case, we offer a new resource that can be fundamental for future developments of assistive technologies. In the other, we propose a small-vocabulary ASR model to build a communication aid device for people with ALS.

The reason led us to work on small vocabulary ASR is that, unfortunately, only few and limited dysarthric speech corpora are available. At the same time, ASR systems are usually based on neural networks that require a large amount of data and, roughly speaking, follow the rule for which “more data imply better performance”. This is especially the case for large vocabulary tasks. Such a behavior has been shown in [159] in which TED-LIUM 3 outperforms the model trained on TED-LIUM 2 dataset by doubling the amount of data. Similarly, VoxCeleb 2 augmented the number of utterances from 100000 to one million, and this allowed to increase the ASR performance [160].

If from one side working on small-vocabulary ASR reduces the problem complexity and the amount of required data, on the other side it forces us to choose a limited number of utterances that can be recognized. Therefore, in this chapter we focus on Voice Command Recognition systems for specific tasks that aim at improving the quality of life of people affected by dysarthria.

Depending on the location of the central and/or peripheral nerve damage, an individual with dysarthria may manifest a variety of motor impairments including weakness, slowness, incoordination, or altered muscle tone. Generally, Augmentative and Alternative Communication (ACC) devices are recognized as appropriate treatment for these people who experience complex communication needs and severe motor abnormalities. Indeed, the use of support technologies can make possible the interaction with the external world that would otherwise result difficult and frustrating. In [161], the authors investigated the impact of these communication devices on the quality of life of people affected by ALS and showed that ACCs have a positive influence on depression and psychological distress.

In the use of assistive technologies, we can identify two main utilities. From one side, they can overcome the poor intelligibility of the speech by synthesizing normal speech or by converting the incomprehensible speech into a readable text. On the other side, based only on the voice, they can provide a strategy alternative to the motor commands to perform some tasks, such as typing a message, using a computer or mobile devices. During the past decades, a wide range of ACC strategies and technologies has been implemented [38–41]. However, these supports do not cover all the individual needs and, sometimes, they turn out to be inconsistent with the changes in the impairment of subjects.

In the following, we introduce the aforementioned two projects whose final purpose is to create ACC devices based on Voice Command Recognition. Specifically, in Sec. 4.1, we propose the EasyCall project in which we collected a dysarthric speech dataset of vocal commands that can be employed in the future to develop a Contacts smartphone application. In Sec. 4.2, we introduce the AllSpeak project in which we realized a speech-to-text mobile application for people with ALS able to recognize their fundamental needs (e.g., “I am hungry”), even when their speech is difficult to be understood by human listeners.

4.1 EasyCall project

In this section, we introduce the EasyCall project that aims at collecting a database of Italian vocal commands recorded from dysarthric subjects. These commands are designed in order to be employed, in the future, for developing a voice command-based Contacts application for smartphone.

The reason underlying the choice of this type of commands derives from the patients’ motor impairments and the consequent difficult they may have in using a smartphone, whose primary utility is being a tool to communicate with the family or the caregivers. Indeed, subjects affected by dysarthria can present one or more sensorimotor problems such as paralysis, involuntary movements, incoordination, excessive or reduced muscle tone [162]. For example, Parkinson’s disease (PD) is characterized by several motor control abnormalities, including bradykinesia (slowness of movement), tremor, dystonia. Amyotrophic Lateral Sclerosis (ALS) involves muscle weakness, fasciculations, spasticity. Often, residual speech is the last and most effective way of interacting with the external world.

EasyCall project focuses on the specific case in which the patient tries to make a call but damages in the movement of the fingers or the arm make this task hard. We considered this scenario as sometimes calls can be the only way to communicate. Indeed, dysarthria typically affects older people who may live alone at home. Also, patients have to regularly go to the hospital and need to communicate with their family.

In the following, we detail the data acquisition process and the resulting corpus. Finally, we discuss the future perspectives, including the use of such a dataset to train a voice command recognition and develop a mobile phone application allowing to make calls and manage phone contacts. To the best of our knowledge, the collected dataset represents the largest Italian dysarthric corpus to date.

4.1.1 Data collection

Participants We recorded utterances from both healthy and dysarthric speakers. The inclusion criteria for the latter group were:

- age > 18;
- dysarthria deriving from Parkinson's Disease, Huntingon's Disease, Amyotrophic Lateral Sclerosis, peripheral neuropathy, myopathic or myasthenic lesions.

Among them, we excluded patients that were not able to carry out the required task. In particular, we did not included individuals that also had one or many of these syndromes:

- aphasic syndromes;
- dementia;
- intellectual disability.

For each subject, an expert classified the type and the severity of the dysarthria by using the Therapy Outcome Measure (TOM).

Experimental design We collected speech recordings of commands related to the task "make a call", with the idea of creating a dataset that could be exploited, in the future, for building a smartphone application able to

- save new contacts;
- type and call phone numbers;
- provide additional options (e.g., rapid/favorite contacts, speakerphone, etc.).

In order to collect a corpus suitable to build an assistive technology in real-life contexts, the choice of the commands results to be particularly important.

One crucial remark is that a Contacts application typically contains a very large number of proper names, as well as frequent contacts such as "Home", "Mum", "Dad". Recording all of them would be unfeasible. Thus, in a hypothetical spoken command-based smartphone application it would not be allow to directly say, for instance, "call home". Rather, the task would be accomplished through a step-by-step procedure. A reasonable sequence of commands could be "Start the Contacts app", "Scroll down to the letter H", "Go down" (repeated until the desired contact is reached), "Call", "End the call".

Moreover, we cannot limit to consider commands of common use. The design of such an experiment, including the list of utterances to record, needs to take into account that a speech-impaired person may use different words to express the same concept due to difficulties in pronouncing some words or, also, we have to consider that the average age of dysarthric subjects is usually high.

In order to record plausible commands, we split our experiments into two phases detailed below.

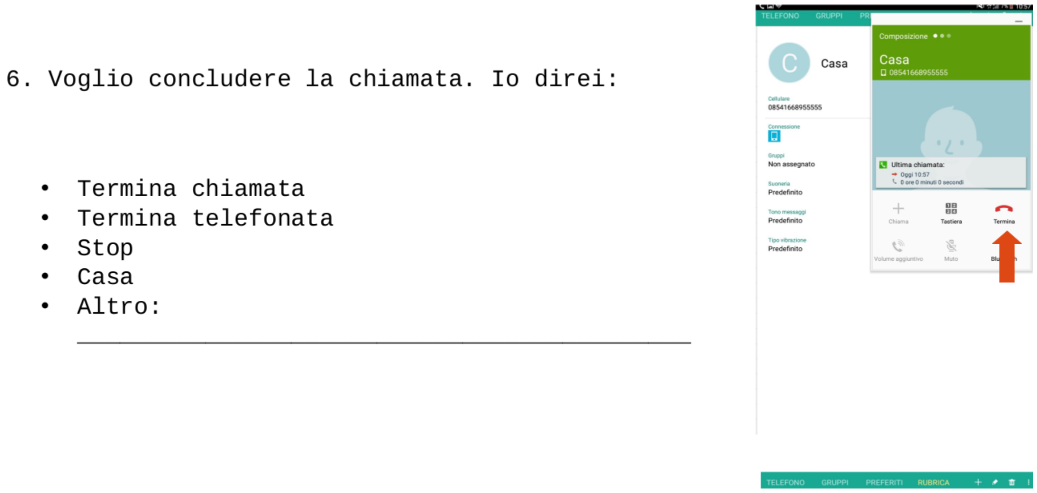
1. **Extrapolation of a reasonable commands list used by dysarthric subjects.**

   We initially designed this stage in order to record spontaneous speech in a controlled setting. On one side, the purpose is to let patients indirectly suggest us the list of feasible commands. On the other side, we forced the speaker to achieve the task step by step.

   Specifically, we equipped the subject with a smartphone controlled by us through a screen recording application. We asked the patients to simulate the task of calling the contact "home". They could use only vocal commands and they were not allowed to say the word "home". This constraint avoids patients to use the command "call home" and obligates them to a sequence of commands, from the starting of the application to the end of the call.

   This restriction also provides a structure to the task as it directs the spontaneous speech to a chain of commands, corresponding to required task steps. In the example given above, the task could be performed in only five steps ("Start the Contacts app", "Scroll down to the letter H", "Go down", "Call", "End the call"). Having a low number of commands has two enormous advantages: 1) it simplifies the task for the patient; 2) it allows us to record multiple examples of the same command. Indeed, as patients tend to get tired soon, the time of the recording sessions is limited. At the same time, due to the poor intelligibility of the dysarthric speech, it is more convenient to have several recordings of the same utterance rather than very few recordings of many utterances.

   Unfortunately, the understanding of the experiment turned out to be not immediate. This was mainly caused by the high average age of the patients that were not very familiar with the use of smartphones. Therefore, we simplified the task by administering a survey that consists of ten questions covering all the steps to make a call and some additional options (e.g., add a contact to the list of the favorite ones). For each question, we suggested 3 reasonable commands, 1 completely unrelated command and the option “other” for which the subjects were asked to give their own answer. For a better understanding of the survey, we also provided a figure beside each question illustrating a smartphone Contact application and the task step asked to perform in the question. Figure 4.1 shows one of the questions, and its illustration, present in the survey.



```
6. Voglio concludere la chiamata. Io direi:

  •  Termina chiamata
  •  Termina telefonata
  •  Stop
  •  Casa
  •  Altro:
     ____________________________________________
```

Figure 4.1: We show one of the questions of the survey: the subject is asked to conclude the call. The first two options are two Italian variants of “end the call”. The third is “stop”. The fourth is a misleading command (“home”). The last option is “other”.

2. **Audio recording.**

   We designed a Smartphone App that showed the sentences the subject were asked to read and, simultaneously, recorded the speech. To avoid tiring the patient, we subdivided the recording stage in sessions. We developed the App in order to allow the participants to take a break within a session, and to ensure that each session contained all commands. The audio was automatically saved in the .wav format on the smartphone device in an anonymous way. We chose to record the audio through the microphone of a smartphone as the final ideal goal of the project is the development of a smartphone application.

The dataset To understand the effectiveness and the importance of the first stage of our experiment, we asked healthy people to fill the same survey and we observed a mismatch in the answering. For example, 100% of healthy subjects selected the first answer in the question showed in Fig. 4.1, while only the 50% of the patients preferred it. Table 4.1 reports the highest scored answers in the survey. As we can see, only three times the answers given by control and dysarthric subjects coincide.

Based on the surveys, we created (and then updated) the list of commands resulting in a final list of 67 sentences. Among these, we have 37 commands (words or sentences related to the task of interest) and 30 non-commands. The last ones can be words near or inside commands (e.g., the non-command “Contacts” is contained in the command “Start contacts”) or sentences phonetically close to the commands (e.g., the non-command “Tra”, that means “between”, is close to the command “Tre”, i.e. “three”). These can be employed to build a more robust Voice Command Recognizer that better discerns between targets that sound similar. Table 4.2 shows the complete list of both commands and non-commands. As they are in Italian, we provide their English translation in brackets.

We recorded the speech commands from 21 dysarthric speakers and 20 healthy speakers, each of whom performed from 3 to 7 sessions. In a session, the speaker repeats each command once and the key command ”start the contact application” three times for a resulting number of 69 recordings per session. The total number of recording is 16086, of which 8283 are from healthy speakers and the remaining 7803 from dysarthric ones.
In Table 4.3 we detail the data we collected so far by reporting the number of recorded sessions and speech for each speaker, and the type and the severity of dysarthria in patients.

Task	**Most voted choice by healthy speakers**	**Most voted choice by dysarthric speakers**
Start the Contact App	Rubrica *Contacts*	Apri rubrica *Start contact*
Scroll down	Vai giù *Scroll down*	In basso *Down*
Make a call	Chiama *Call*	Chiama *Call*
Select the speakerphone	Attiva vivavoce *Activate speakerphone*	Alza volume *Increase the sound*
End a call	Termina chiamata *End call*	Termina chiamata (50%), Stop (50%) *End call* (50%), *Stop* (50%)
Save a contact as favorite	Aggiungi ai preferiti *Add to favorite*	Aggiungi ai preferiti *Add to favorite*
Check the list of favorite contacts	Lista preferiti *List of favorite*	Vai ai preferiti *Start favorite*
Type a new number	Vai a tastiera *Start Keypad*	Vai a tastiera *Start Keypad*
End the Contact App	Chiudi rubrica *End contacts*	Esci da rubrica *Quit contacts*

Table 4.1: EasyCall Survey results. We report the most voted choice by healthy and dysarthric speakers for ten questions. As the survey is in Italian, we translated the answers (in italic). The answers of the two groups match only three times, whereas these differs in the remaining seven cases.

Commands	Apri rubrica (*Start contacts*), Scorri verso il basso (*Scroll down*), Scorri verso l'alto (*Scroll up*), Stop (*Stop*), Sali (*Go up*), Scendi (*Go down*), Seleziona (*Select*), Aggiungi ai preferiti (*Add to favorites*), Deseleziona (*Deselect*), Rimuovi (*Remove*), Si (*Yes*), No (*No*), Indietro (*Back*), Aggiungi (*Add*), Vai nella rubrica (*Go to contacts*), Vai al registro chiamate (*Go to call history*), Chiama (*Call*), Chiama emergenza (*Call emergency number*), Attiva vivavoce (*Activate speaker*), Disattiva vivavoce (*Disable speaker*), Termina chiamate (*End call*), Zero (*Zero*), Uno (*One*), Due (*Two*), Tre (*Three*), Quattro (*Four*), Cinque (*Five*), Sei (*Six*), Sette (*Seven*), Otto (*Eight*), Nove (*Nine*), Cancella (*Delete*), Cancella tutto (*Delete all*), Chiudi rubrica (*End contacts*), Esci da rubrica (*Leave contacts*)
Non-commands	Rubrica (*Contacts*), Preferiti (*Favorites*), Sezione (*Section*), Vivavoce (*Speakerphone*), Chiamata (*Phone call*), Richiama (*Recall*), Tastiera (*Keypad*), Terminare (*End*), Apri (*Start*), Chiudi (*End*), Muto (*Mute*), Zelo (*Zeal*), Nave (*Ship*), Mali (*Pains*), Tra (*Between*), Scesi (*Descended*), Bue (*Ox*), Sotto (*Below*), Sopra (*Above*), Salve (*Saved*), Muovi (*Move*), Raggiungi (*Reach*), Cella (*Cell*), Top (*Top*), Cancella contatto (*Delete contact*), Vai alla pagina principale (*Open main page*), Fai una telefonata (*Make a call*), Chiudi applicazione (*End application*), Nuovo contatto (*New contact*), Chiama ultimo numero (*Call last number*)

Table 4.2: Final list of recorded utterances, drawn up after the survey results.

Speaker	Type of dysarthria	TOM rating	N. Sessions	N. wav files
FC01	-	-	6	396
FC02	-	-	6	396
FC03	-	-	6	396
FC04	-	-	6	396
FC05	-	-	7	483
FC06	-	-	6	414
FC07	-	-	6	414
MC01	-	-	6	396
MC02	-	-	6	396
MC03	-	-	6	396
MC04	-	-	6	396
MC05	-	-	7	462
MC06	-	-	6	396
MC07	-	-	7	462
MC08	-	-	6	414
MC09	-	-	6	414
MC10	-	-	6	414
MC11	-	-	6	414
MC12	-	-	6	414
MC13	-	-	6	414
F01	paretic	1	5	330
F02	paretic	3	6	396
F03	paretic	1	6	396
F04	paretic	1	6	414
F06	paretic	1	6	414
F07	paretic	1	6	414
F08	paretic	1	6	414
M01	extrapyramidal	2	6	396
M02	paretic	1	6	396
M03	paretic	2	6	396
M04	paretic	1	6	396
M05	paretic	3	6	396
M06	paretic	3	6	396
M07	paretic	3	2	132
M08	paretic	1	6	396
M09	cerebellar	1	6	396
M10	paretic	1	6	414
M11	paretic	4	4	276
M13	paretic	1	6	414
M14	pyramidal	4	3	207
M15	paretic	1	6	414
Total	-	-	239	16086

Table 4.3: EasyCall Dataset. The mismatch between the number of wav files, with the same number of sessions, for some speakers depends on the fact that we updated the command list during the experiment time by adding new commands.

4.1.2 Discussion and future work

To the best of our knowledge, this is the richest Italian dysarthric speech corpus. Note that this project is still ongoing and, hence, this database is destined to increase. We did not evaluate the dataset on general ASR systems (e.g., Google Speech API, IBM, Microsoft) for lack of time but we left it for future work.

Unfortunately, the acquisition of dysarthric speech is a long process due to several issues. As well as the red tape, we have to take into account the difficulties the patient can encounter. For instance, it is a good practice to ask the patient to take part in the recording session the same day in which she has the medical visit at the hospital, in order to avoid her an additional trip. Often, the recording sessions have to be interrupted as the patient is too fatigued. These adversities usually limit the number and the duration of the recordings, resulting in small speech data.

Indeed, a similar database has been collected in British English. The homeService corpus [163] has been gathered as part of the homeService project that aims at helping dysarthric individuals to operate their home appliances using voice commands. This dataset has the merit of containing real home environment recordings, as well as voice commands to control the home appliances. However, it consists of audio of only five dysarthric speakers.

Popular dysarthric speech corpora in America English are the TORGO dataset [14], the Nemours corpus [164], and the Universal Access (UA) speech [165]. The first, already introduced in Chapter 3, includes 5980 audio recorded from 7 healthy speakers and 2762 utterance recordings of 8 dysarthric speakers. The Nemours database contains 814 short nonsense sentences, 74 sentences spoken by each of 11 dysarthric speakers. To the best of our knowledge, the UA-Speech database is the largest corpus of dysarthric speech in American English. It is a collection of 541 read speech recordings from 19 individuals with cerebral palsy. The prompt words include: three repetitions of the first ten digits, three repetitions of 26 radio alphabet letters, three repetitions of 19 computer commands, common words form the "Grandfather Passage" and uncommon words from phonetically balanced sentences (TIMIT [76]) one time each.

The corpus we collected consists of 16086 commands recorded from 20 healthy subjects and 21 dysarthric speakers. Thus, it represents one of the largest databases of speech recorded from subjects with dysarthria. We firmly believe this corpus can provide a fundamental resource for developing assistive technologies beneficial to individuals with dysarthria.

This dataset may be exploited to train a voice command recognition system that allows dysarthric individuals to control a Contact mobile application by voice and make calls in an easy way. Moreover, the data recordings include non-commands phonetically close to commands (e.g., "Tre" is close to "Tra") or near/inside the commands (e.g., "End" is included in "End call"). The use of these audio tracks can improve the robustness of the grammar model.

Further, one may use it not only with the purpose of developing a Contacts mobile application, but also to pre-train a network for a different goal. Indeed, when the dataset size is limited, a common strategy to improve the ASR system performance is pre-training the network model by a larger dataset. Due to lack of dysarthric recordings, this step is usually achieved by using healthy speech datasets. Pre-training a network with dysarthric data would bring a more representative model in which the training distribution is closer to the distribution of the data of interest.

4.2 AllSpeak project

In the AllSpeak project, we aim at developing a device to assist people affected by Amyotrophic Lateral Sclerosis (ALS) in the interaction with other persons. Specifically, we designed an Android application for both smartphones and tablets that supports ALS patients in the verbal communication, especially when speaking becomes a strenuous task and their voice intelligibility almost vanishes. The App is based on a Voice Command Recognition system that allows patients to communicate, through their residual speech abilities, their basic needs. The development of the Android App has been carried out by a member of the project team, Alberto Inuggi.

4.2.1 AllSpeak App

The AllSpeak App is a hybrid App developed with the Ionic 1.X framework for the Android 6.0 platform. All the speech processing and recognition modules are implemented within a custom multi-threaded Cordova Plugin. The latter is composed by the following modules, each running on its own independent thread:

- audio acquisition (INPUT), it extracts speech from the smartphone's microphone and sends it to the next module;
- voice activity detection (VAD), it sends the speech segments to the FE module if speech activity is recognized;
- spectral features extraction (FE), it computes the spectral features and sends them as concatenated feature vectors to the TF module;
- inference on the tensorflow loaded model (TF), it translates the speech command into text.

Once recognition is activated, these four processes run in parallel. The VAD module sends "active samples" up to 400 ms once the speech activity is finished. If the duration of the detected speech segment is above a predefined threshold (500 ms in our case), then it is considered as a command and it is sent to the TF module which will estimate the command once given the spectral features' vectors of the segment. This four-threads approach optimizes the recognition process, since the to-be-inferred features are already present in the TF module when the VAD module decides that a new command has been pronounced by the App user.

4.2.2 Voice Command Recognition

The TF module, introduced in the previous section, is based on deep neural networks (DNNs). The first version of our algorithm running on the App was based on a feed-forward DNN[42]. The decoder simply averaged the spoken command posterior probabilities outputed by the DNN at each speech frame and selected the command with the highest posterior. The DNN was trained on non-dysarthric speech and then adapted on the dysarthric speaker of interest by adding one or many hidden layers to the speaker-independent (SI) model. This recognizer had an averaged command error rate of 32.7% on a subset of 8 speakers.

With the aim of improving the recognition performance, we here explore an alternative strategy to learn a more efficient SI model: we train a sequence-to-vector BLSTM on both dysarthric and healthy speech recordings. To compensate to the mismatch between the SI model and the probability distribution of the speaker of interest, we also perform supervised speaker adaptation.

Dataset The dataset consists of speech recordings from 13 healthy (or control) speakers and 17 speakers affected by Amyotrophic Lateral Sclerosis (ALS). The control dataset includes 23 commands, each of them is repeated from 8 to 10 times by every speaker (Tab. 4.4). The ALS patients recorded the same commands and repeated each one of them from 4 to 10 times, depending on patient's medical condition (Tab. 4.5). From each speech signal, we extract 13 Mel Frequency Cepstral Coefficients (MFCCs) with a frame Hamming window of 25 ms length shifted every 10 ms. We also consider temporal delta and acceleration coefficients. The final extracted vectors result in 39-dimensional features. The Speaker-independent model is trained on the *source* dataset, consisting of the control dataset and all the dysarthric speech recordings excepted the ones recorded from the dysarthric speaker of interest (called *target*). Speaker adaptation is finally performed by using only the small target speaker data, split into training and testing subsets.

Control speaker	MCO1	FC01	FC02	MC02	MC03	FC03	MC04	FC04	MC05	FC05	MC06	FC06	Total
N. commands	142	175	245	300	249	226	129	146	223	161	230	161	2387

Table 4.4: AllSpeak: Control speech dataset

Dysarthric speaker	M01	M02	F01	F02	M03	M04	M05	F03	M06	M07	M08	M09	F04	M10	M11	M12	M13	Total
N. commands	119	117	57	125	45	125	126	115	115	125	125	119	114	125	125	125	55	1857

Table 4.5: AllSpeak: Dysarthric speech dataset

Model In order to capture long-time dependencies, we adopt a Bidirectional Long Short-Term Memory (BLSTM). Typically, in speech recognition, both recurrent and feed-forward networks are trained as frame-level classifiers. As a consequence, the alignment between audio and transcription sequences has to be determined in order to have a target for every frame. These alignments are usually provided by a Gaussian Mixture Model – Hidden Markov Model (GMM-HMM) system trained with the Baum-Welch algorithm. However, a good alignment of impaired speech may not be feasible, and that can have catastrophic consequences on the (frame-level) training of neural networks (as labels would be very noisy). To address this issue, we train the BLSTM as a sequence-to-vector model [166] to perform command recognition over 23 classes. Sequence-to-sequence methods allow to train a network by taking in input a sequence of length T and giving as an output the correspondent sequence of length T', where T and T' are not necessarily the same. In our case, the output sequence is a command and, therefore, $T = 1$. The underlying idea is very simple: an encoder (or reader) BLSTM processes the input sequence and emits a fixed-size context variable C, which represents a summary of the input sequence. A decoder (or writer) takes as input the context C and generates the output sequence. Usually, the final hidden state of the encoder is used to compute C. In terms of probability, the sequence-to-vector architecture maximizes the probability of the command, given the whole acoustic sequence, $p(y|x_1, \cdots, x_T)$.

Once trained the speaker-independent model, we perform supervised speaker adaptation to reduce the mismatch between the acoustic model and the testing speaker. More precisely, after training the network, we add a feed-forward linear layer atop the input. We train the new layer, freezing the other ones, on the adaptation data.

Experimental setup We evaluate the sequence-to-vector BLSTM on the AllSpeak dataset. In particular, we test five patients and two control speakers in order to cover the whole range of dysarthric degrees (on the TOM scale). From the speech of these speakers we extract the adaptation data and the testing data. For all the experiments, we use the BLSTM network with 5 hidden layers and 250 units per layer. We set the initial learning rate to be 0.01, and we exponentially decayed the learning rate by a factor of 0.7, every 3000 steps. Our model is trained to minimize the cross entropy (within the sequence-to-1 paradigm), by using the momentum optimizer with momentum equal to 0.9. We also clip the gradient to avoid the vanishing/exploding gradient problem. Cross-validation is employed to get the best number of training epochs.
A linear layer at the bottom of the BLSTM architecture has been added in a second phase to perform speaker adaptation. We initialized the weight as the identity matrix and the bias as the zero vector.

Results Table 4.6 shows the command error rate (CER). As expected, the error is lower on the control speakers. Surprisingly, the sequence-to-vector BLSTM achieves a good performance even in presence of dysarthric speech, with a minimum error of 4% on the speaker M09. In every case, the error is reduced (or remains equal) after the speaker adaptation phase. In the best case, adaptation provides an error reduction from 71.7% to 21.7%. Note that the averaged error rate is not referred to all speakers but only to dysarthric ones for which the TOM is available. The Speaker-Adapted model always outperforms both the Speaker-Independent model and the model implemented in the first App version [42], that was based on feedforward neural networks. As we can see, we obtain a CER reduction from 33.9% to 14.2% by applying the speaker-adaptation phase to the BLSTM trained model. Surprisingly, we have a CER reduction of 64% over the first AllSpeak App version.

Speaker	TOM	First App Version [42]	BLSTM-SI model	BLSTM-SA model
M02	0	-	7.0	**7.0**
F02	0	-	8.0	**1.3**
F01	NA	-	44.4	**25.9**
M02	1	47.8	25.0	**18.2**
F03	1	30.4	71.7	**21.7**
M01	3	36.4	34.8	**13.0**
M09	3	47.8	4.0	**4.0**
Average	-	40.6	33.9	**14.2**

Table 4.6: CER (%) provided by the feedforward DNN model implemented in the first App version, the BLSTM-based speaker-independent model and BLSTM-based speaker-adapted model. We report the results for different level of dysarthria severity. NA stands for Not Available.

4.2.3 Discussion

In this work, we adopt a recurrent neural network (NN) rather than a feedforward NN, as proposed in the first version of AllSpeak App. Specifically, we train a Voice Command Recognition as a sequence-to-vector BLSTM. This has two main advantages over the feedforward NN. First of all, BLSTMs are known to outperform feedforward networks in speech tasks as they can capture long-term dependencies and better model the co-articulation effects. Secondly, adopting a sequence-to-vector architecture avoids to pass through state alignment step. This is a risky procedure as the alignment of dysarthric speech may be unfeasible or inaccurate. Clearly, wrong labels would cause a misleading training and a completely inaccurate recognition system.

To move a step forward, we insert an additional linear layer at the bottom of the BLSTM to perform speaker adaptation, once the SI model is trained. In this context, speaker adaptation turned out to be fundamental providing an absolute CER reduction of 19.7 % and 26.4 % over the BLSTM and the feedforward NN, respectively. The great improvement we obtained points out a large gap between the training distribution and the distribution of the speaker of interest. Indeed, the speaker differences are here accentuated by ALS severity. Patients can show a variety of different impairments, from some speech disruptions to an almost vanish speech intelligibility. Nevertheless, results also suggest us that the relationship between a speaker and the SI representation can be model by a linear function. Probably, this comes from the homogeneity of the dysarthria type as all the considered patients are affected by ALS and, hence, their speech is characterized by common impairments.

Chapter 5

Conclusions

In this thesis, we investigated three ASR-related problems that involve different research lines. These range from the integration of the speech production information to the ASR system, inspired by neurophysiological studies [20, 45, 46], to multi-source domain adaptation problem [95–101], to a more technological approach finalized to the production of healthcare tools. Although these approaches implement different points of view, they are motivated by the common goal of improving ASR systems and creating new assistive technologies for people affected by dysarthria.

We focused on ASR systems for dysarthric individuals as ASR-based devices can result particularly beneficial for them. Indeed, dysarthria often presents an almost vanishing speech intelligibility or motor control impairments and, hence, these technologies represent the only possibility to interface with other persons or machines.

Dysarthria is a widespread motor impairment and even if its incidence in the world is not fully known, it is a frequent consequence of several neurological conditions. It is estimated to affect approximately 70%–100% individuals with Parkinson's disease [81, 167, 168], and 25%-50% individuals with multiple sclerosis [167, 169, 170]. Dysarthria can be observed as an initial sign in up to 30% of individuals with Amyotrophic Lateral Sclerosis, and it is present in almost all individuals in later stages [171–173]. It is also associated with Stroke [174–177] and Traumatic Brain Injury [178–182]. Moreover, these disorders are often characterized by motor control abnormalities [162] (e.g., involuntary movements, paralysis) that impede an easy motor interaction with devices and to conduct a normal life. Therefore, devices based on vocal commands may support patients in connecting with other people and using machines.

If from one side the demand of ASR-based technologies is increasing, on the other side the supply is scarse. Indeed, traditional ASR models fail in presence of dysarthric speech. For instance, we tested two of the best speech-to-text systems, i.e. Google Speech API and IBM, on a subset of the TORGO dataset [14]. They obtained 81% and 96% WER, respectively, while the human error is 30%. Other studies [7–10] also confirm our finding. This gap between human and machine errors highlights the lack of systems able to convert the dysarthric voice into text.

The underlying reason is that dysarthria adversely affects the intelligibility and naturalness of speech. For example, an individual with dysarthria may have some impairments in respiration, bringing to short phrases, reduced loudness or forced expiration/inspiration. Articulation disruptions may lead to imprecise consonants and distorted vowels, irregular articulatory breakdown and articulatory blurring. The patient's speech may also be altered in the prosody, showing an abnormal speaking rate (too fast/too slow/variable), excessive or equal stress to all syllables, prolonged intervals or inappropriate silences. Other common impairments are aberrant pitch level (too low/too high) and voice quality (e.g., roughness or hoarseness), voice tremor and stoppage.

However, the speech characteristics vary by dysarthria type and severity. The ideal ASR system should be trained on datasets covering most of these speech impairments. Unfortunately, collecting such a corpus is pretty hard due to the difficulties in both obtaining the legal authorization from the hospitals for recording the speech and having long sessions of recording, as the patients tend to get tired soon. As a consequence, the existing dysarthric speech corpora are very limited and ASR systems poorly generalize to new datasets or speakers.

As mentioned at the beginning, to overcome such an issue, we individuated three possible strategies. The first consists in exploiting additional features, as well as the acoustic ones. As dysarthria primarily involves motor impairments, we proposed to take advantage of articulatory information for improving the ASR performance. This allows us to directly model the vocal tract disruptions, rather than the consequent complex acoustic effects. The integration of speech production knowledge in a speech recognition system is also motivated by the Motor Theory of Speech Perception (MTSP) [19], stating that the perception of speech involves the perception of articulatory features, and neurophysiological evidences [20] confirming that the activity of the motor cortex contributes to the speech perception.
The second method we investigated attempts to leverage other larger datasets, called source domains, to learn a classifier for the dataset of interest called target domain. This is known in machine learning as multi-source domain adaptation problem. In particular, we developed an algorithm based on the Optimal Transport (OT) Theory [32, 119, 123–125]. This estimates the similarity between the source and target domains and learn a target classifier using only the most similar source datasets.
In the third and last approach, we reduced the recognition vocabulary of the ASR system to a list of commands focused on a specific task.

It is crucial to remark that the proposed approaches are independent but not in contrast each other. Therefore, one may employ all of them in the same model. For instance, performing domain adaptation on speech production knowledge may be particularly advantageous. Indeed, articulatory representation is more compact than the acoustic one and, hence, the adaptation process could be faster and would require less amount of data. In the following, we summarize the proposed methods and the obtained results.

Speech production knowledge for ASR

As said, the purpose of this approach is the integration of articulatory features (AFs) in the ASR model, in addition to the acoustic representation. However, an underlying problem in using the AFs is that articulatory recordings are difficult to collect and laborious to pre-process. As a consequence, often we do not have access to articulatory information or only a small articulatory dataset is available for training. A common procedure to recover the AFs in a supervised framework is learning an acoustic-articulatory mapping, also known as Acoustic Inversion (AI) map [22–24, 63–65].

In Chapter 2, we employed two types of phonetic features in addition or substitution to acoustic information in the AI map with the aim of improving its generalization across speakers and across datasets. Further, we confronted the case in which we do not have access to motor measurements and proposed autoencoder-based methods to synthesize AFs from phonetic and acoustic information.

The two phonetic features we adopted are the Linguistic Features (LFs) and the Statistical Features (SFs). The first features come from linguistic observations, such as the synchrony constraints on pairs of AFs. The second ones are the result of a statistical procedure we proposed to represent the average configuration of the vocal tract of a speaker during the production of each phoneme. Both feature types can be extracted from a look-up table, given the phonetic annotation, and contain raw articulatory information corresponding to the phoneme sequence. In the framework in which the AF measurements are not available, we exploited the LFs/SFs to capture the raw motor information and the acoustic features to embed complex phenomena, such as the coarticulation effects.

We validated the accuracy of the generated AFs in two training-testing conditions: matched (the generalization is within the same dataset, across speakers) and mismatched (the generalization is across datasets). In both scenarios, results suggested that the use of LFs/SFs outperforms the standard AI both if substituted or added to the acoustic features. Moreover, the employed deep learning methods modulate the LFs/SFs generating more accurate AFs. What was unexpected and surprising is that, in the supervised and mismatched condition, the use of the MFCCs deteriorates the reconstruction performance. Indeed, the SFs-to-AFs mapping outperforms the AI map in which both SFs and MFCCs are used as input, although the latter map disposes of more information. This is due to the strong speaker dependency of MFCCs (despite their per-speaker normalization) and stresses out the effectiveness of the proposed statistical features. We found the SFs-to-AFs mapping being the most effective method in the mismatched training-testing condition over the standard approach and all the other proposed methods. Therefore, we adopted it to synthesize the AFs for only-speech corpora.

It is fundamental to recall that the LFs and SFs are phonetic features and, hence, require the access to the phone labels. In an ASR task, the phonetic transcription is available only during the training and, consequently, we cannot employ the LFs/SFs to synthesize the articulatory features. For this reason, we firstly proposed an articulatory ASR that uses the articulatory information only during training. Specifically, we proposed to adopt the synthesized AFs as secondary target in the ASR training.

Alternatively, we proposed to add an intermediate step between the SFs-to-AFs mapping and the integration of speech production information in the ASR. More precisely, we employed the output of this mapping (i.e. the estimation of the AFs) as target for a second mapping in which the input is given by the acoustic features. Therefore, we learn an audio-to-estimated AFs mapping that does not require the use of phone labels and, hence, can be used in the testing phase. Finally, we integrated the AFs synthesized by this last mapping into the ASR as additional input to the acoustic features.

We tested our approach on two well known speech datasets, i.e. TIMIT and CHiME-4. Specifically, the model was trained to perform phoneme and senone classification on first and the latter dataset, respectively. Results on these corpora showed that using the synthesized AFs always improves the recognition performance. In particular, preliminary studies on TIMIT reported a relative FER reduction of 1.4% and 6.2% by using the AF as secondary target or additional input, respectively. Even though both proposed approaches outperformed the audio-only based model, the latter strategy turned out to be the most effective one. This was also confirm on the CHiME-4 corpus for which concatenating the synthesized AFs with the acoustic input reduces the WER from 22.77% to 21.54%.

We firmly believe the proposed approach can be employed to improve the ASR for dysarthric speech. Indeed, the use of articulatory knowledge may be particularly beneficial in this context for two reasons. Firstly, dysarthria is a motor disorder involving vocal tract disruptions and, hence, exploiting the articulatory features seems a natural choice. Secondly, as dysarthric corpora are usually very limited, ASR models cannot adequately capture the inter-speaker variability. Articulatory features offer a compact representation through which complex surface phenomena can be described by constraints in the vocal tract dynamics, as shown in [183] for pronunciation modeling. This property may also be exploited for speaker adaptation, where the small amounts of data available may be not sufficient to represent all the target labels.

Further, articulatory-acoustic corpora recorded from dysarthric individuals are extremely rare and, hence, the majority of studies [22–24], that require motor measurements, cannot be employed in this context. On the contrary, our approach relies on the Statistical Features that can be extracted from a look-up table by only using the phone labels. These can be further adapted to the speaker (and dysarthria) characteristics by leveraging the acoustic information in the proposed autoencoder based methods.

Multi-source domain adaptation

In Chapter 3, we address the multi-source domain adaptation (MSDA) problem whose purpose is learning a classifier for an unlabelled target domain by taking advantage of some labelled source domains.

We computed the joint empirical distributions of the source datasets (X_s, Y_s) and considered the proxy joint distribution for the target domain, as proposed in [85]. Specifically, this is computed by replacing the labels with the prediction of the classifier $f(X)$. We proposed the Multi-Source Domain via Weighted Joint Optimal Transport (MSDA-WJDOT) that minimizes the Wasserstein distance between a convex combination of the source joint distributions and the target proxy joint distribution. MSDA-WJDOT simultaneously learns the optimal source weights $\boldsymbol{\alpha}$ and target classifier f. Intuitively, this algorithm looks for the most similar source distributions and "imitate" them by f.

As stated in the "impossibility theorem" [127], the adaptation is not possible if the target distribution is too different from the source one. Following this statement, MSDA-WJDOT algorithm discards the sources that are not close to the target and promotes the adaptation to the most similar ones. The characteristic of our approach is that it provides a measure of domain relatedness allowing to select only the important sources and to interpret the data.

To support the validity of the proposed approach, we derived a generalization bound on the target error in which the first term is the Wasserstein distance minimized by MSDA-WJDOT, and the other terms can be assumed small or controlled. We then conducted a study about the stability and the convergence of the algorithm. We found that the algorithm is not affected by the initialization of the model parameters as the loss function always converges. Then, we observed a fast convergence of the weights $\boldsymbol{\alpha}$, even for an increasing number of sources. This suggests that MSDA-WJDOT is able to rapidly select the only relevant source domains.
Further, we evaluated the proposed method on simulated data to confront both domain and target shift problems. We tested it on several conditions (e.g., small/high number of sources, small/large source/target datasets), and we always found that our approach outperforms the state-of-the art methods.

Finally, we applied MSDA-WJDOT to the case of interest: MSDA for dysarthric speech. In particular, we considered two interesting applications. The first is domain adaptation for dysarthria detection. Here, we employed several noisy dysarthric datasets as source and target domains. This scenario is pretty common in real life as speech recordings from dysarthric speakers usually take place at the hospitals that are not equipped of designed area to record in absence of noise. We evaluated our model on four target domains, corresponding to different types of noise. We obtained an average relative accuracy improvement of 4.1 % and 0.7 % over the Baseline and the best competitor method, respectively.
The second scenario we confronted is the dysarthric speaker adaptation in a spoken command recognition system. Specifically, we assumed to have access to healthy and dysarthric speaker datasets. We employed MSDA-WJDOT to leverage these multiple labelled datasets and learn a voice command classifier for an unlabelled dysarthric speaker dataset. The surprisingly results showed that MSDA-WJDOT reduces the

Command Error Rate (CER) of 21 % over the Baseline, outperforming also the implemented extensions of JDOT [85].

The strength of this method is that it selects only the most suitable source domains for the adaptation by assigning a score to them, based on their similarity with the target domain. We found that this distribution similarity reflects the closeness in the speech characteristics of the speakers. Indeed, when the target speaker is dysarthric MSDA-WJDOT assigns higher scores to dysarthric speakers, whereas the target speaker is healthy the source speakers with higher weights are healthy too. This is an interesting and unique feature of our approach, that provides an interpretable closeness measure. We took advantage of it to move a step forward.

By summing the weights attributed to healthy and dysarthric source speakers, we defined the Healthy and Dysarthric Index, respectively, and we investigated the use of these indices to detect dysarthria. We found that, for all dysarthric individuals, the Dysarthric Index was always higher than the Healthy one, whereas the opposite took place for the healthy speakers. This demonstrates that MSDA-WJDOT can also automatically detect dysarthria during the Speaker Adaptation without any additional and specific training of the model.

In future work, the algorithm may be further improved. As mentioned above, for a large number of source domains, $\boldsymbol{\alpha}$ becomes sparse. This means that we can select few sources and preserve the computational cost. Also, one may investigate the other terms in the generalization bound. For instance, one of the terms depends from the ratios $\frac{\boldsymbol{\alpha}}{N_j}$, where N_j indicates the size of the j-th source dataset, that could explode for big N_j values. A regularization on $\boldsymbol{\alpha}$ may be introduced to force small α_j coefficients when the sources are poorly sampled.

As the proposed approach is very general, it may be employed in other real-world applications (e.g., computer vision, sentiment analysis, etc.). However, here we only discuss the applications of interest, i.e. the ones on dysarthric speech. As aforementioned, we used MSDA-WJDOT to perform speaker adaptation providing two scores to the speaker, that are the Healthy Index and the Dysarthric Index. We took advantage of these indices to detect dysarthria, based on which index is higher. The relationship between Dysarthric and Healthy Index may be further investigate in order to have a more accurate dysarthria detection. Indeed, we assumed a Dysarthric Index less than 0.5 indicating the absence of dysarthria, whereas it may be associated to a mild dysarthria.

Moving a step forward, we may study the relationship between the Dysarthric Score and the dysarthria severity. Ideally, we could individuate intervals between 0 (absence of dysarthria) and 1 (severe dysarthria) to assess the dysarthria severity. One plausible example could be $[0, 0.3]$=none, $[0.3, 0.5]$=mild, $[0.5, 0.7]$=moderate, $[0.7, 0.9]$=moderate-severe, $[0.9, 1]$=severe.

Integrating such a procedure in an ASR system may warn the speaker of a disease deterioration in order to act in time with the speech therapy. Once again, we need to stress out that this technology does not need any additional data or training but it may be automatically provide by the speaker adaptation model.

Small vocabulary ASR

In Chapter 4, we focused on small-vocabulary ASR. Even if it can be exploited only for a specific task, this system has the advantage of providing a higher recognition accuracy. Such an approach turns out to be particularly suitable to develop Augmentative and Alternative Communication (ACC) devices. These are technologies to support patients who experience difficulties in communication. We can individuate two main types of ACCs, one enhancing the poor intelligibility of the speech by synthesizing it or translating it into text, one supplying the patient with devices based on vocal commands. Thus, in the first case the main goal is improving the human-to-human interaction, whereas the second is allowing the individual to interface with machines. In this thesis, we focused on both frameworks.

Firstly, we developed an assistive tool for people affected by ALS to communicate their basic needs. Indeed, their speech is often characterized by a very low intelligibility and the communication with other individuals can be exhausting. This work is part of a project, named AllSpeak, that led to the production of a mobile application in which a command recognition system is integrated to recognize basic needs, such as “I am thirsty” or “I am hungry”. We adopted a BLSTM model [184] trained as a sequence-to-vector model [166] in order to avoid the speech alignment procedure, that may fail in presence of dysarthric speech. Also, we integrated a speaker adaption technique by inserting a feed-forward layer at the bottom of the network. Results showed that this architecture achieves an average CER of 14.2 %, while a first Application version (based on feedforward neural network) only reached the 40.6 % of CER.

Secondly, we faced the fact that dysarthria often entails motor injuries, such as involuntary movements, paralysis, tremors, etc. Due to these motor abnormalities, patients may experience strong difficulties in using devices, including the mobile phone that is often the only way to communicate with their family or caregivers. Thus, a mobile application guided by the voice to make calls and manage the phone contacts could be particularly useful for them. With this purpose, we collected recordings from dysarthric and healthy speakers containing commands related to the task of making a call. This collection resulted in the largest dysarthric speech corpus in the Italian language to date.

The collection of this corpus clearly paves the way for the development of a smartphone Contact application based on a Spoken Command Recognition system. Such a system would be trained in order to discriminate among 37 commands. Moreover, this corpus includes 30 non-commands that can be used to make the model more robust to similar sounds. For instance, the non-command “Zelo” (*zeal*) is thought to improve the recognition of the command “Zero” (*zero*).

As known, speaker adaptation is a fundamental step in ASR and, especially, in presence of dysarthric speech. Future work may investigate both the supervised approach exploited in the AllSpeak project and the unsupervised one introduced in 3 for this context.

Bibliography

[1] P. M. Enderby. *Frenchay Dysarhtia Assessment*. San Diego, California: College-Hill Press, 1983.

[2] Yorkston K. M, Beukelman D. R., and Traynor C. D. *Computerized assessment of intelligibility of dysarthric speech*. Tigard, OR: C. C. Publications, 1984.

[3] Enderby P. and John A. *Therapy outcome measures for speech and language pathology*. San Diego, CA: Singular Publishing Group, 1997.

[4] Doyle P. C. et al. "Dysarthric speech: A comparison of computerized speech recognition and listener intelligibility". In: *Journal of Rehabilitation Research and Development* 34.3 (1997), pp. 309–316.

[5] Ferrier L. et al. "Dysarthric speakers' intelligibility and speech characteristics in relation to computer speech recognition". In: *Augmentative and Alternative Communication* 11 (1995), pp. 165–175.

[6] Kim Y., Kent R.D., and Weismer G. "An acoustic study of the relationships among neurologic disease, dysarthria type, and severity of dysarthria". In: *Journal of Speech, Language and Hearing Research* 54.2 (2011), pp. 417–429.

[7] F. Rudzicz. "Comparing Speaker-Dependent and Speaker-Adaptive Acoustic Models for Recognizing Dysarthric Speech". In: Tempe USA, October 2007.

[8] F. Rudzicz. "Toward a noisy-channel model of dysarthria in speech recognition". In: Los Angeles, CA, 2010, pp. 80–88.

[9] F. Rudzicz. "Correcting errors in speech recognition with articulatory dynamics". In: Stroudsburg, PA, USA, 2010, pp. 60–68.

[10] Rudzicz F. "Using articulatory likelihoods in the recognition of dysarthric speech". In: *Speech Communication* 54 (2012), pp. 430–444.

[11] Espana-Bonet C. and Fonollosa J. A. R. "Automatic speech recognition with deep neural networks for impaired speech". In: *International Conference on Advances in Speech and Language Technologies for Iberian Languages*. Springer. 2016, pp. 97–107.

[12] Joy N. M. and Umesh S. "Improving acoustic models in torgo dysarthric speech database". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 26.3 (2018), pp. 637–645.

[13] Bhavik Vachhani et al. "Deep Autoencoder Based Speech Features for Improved Dysarthric Speech Recognition." In: *Interspeech*. 2017, pp. 1854–1858.

[14] Rudzicz F., Namasivayam A.K., and T. Wolff. “The TORGO database of acoustic and articulatory speech from speakers with dysarthria”. In: *Lang Resources & Evaluation* 46 (2012), pp. 523–541.

[15] Steven Davis and Paul Mermelstein. “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences”. In: *IEEE transactions on acoustics, speech, and signal processing* 28.4 (1980), pp. 357–366.

[16] Alex Graves et al. “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks”. In: *Proceedings of the 23rd international conference on Machine learning*. 2006, pp. 369–376.

[17] Alex Graves. “Sequence transduction with recurrent neural networks”. In: *arXiv preprint arXiv:1211.3711* (2012).

[18] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. “Effective approaches to attention-based neural machine translation”. In: *arXiv preprint arXiv:1508.04025* (2015).

[19] B. Galantucci, Fowler. C.A., and M.T. Turvey. “The motor theory of speech perception reviewed”. In: vol. 13. 2006, pp. 361–377.

[20] A. D’Ausilio et al. “The motor somatotopy of speech perception”. In: vol. 19. 2009, pp. 281–285.

[21] F. Rudzicz. “Towards a noisy-channel model of dysarthria in speech recognition”. In: Los Angeles California, June 2010, pp. 80–88.

[22] P. K. Ghosh and S. S. Narayanan. “An subject-independent acoustic-to-articulatory inversion”. In: Prague, Czech Republic, 2011.

[23] Weiran Wang et al. “Unsupervised Learning of Acoustic Features via Deep Canonical Correlation Analysis”. In: *Proc. of ICASSP*. Brisbane, Australia, 2015.

[24] L. Badino et al. “A Speaker Adaptive DNN Training Approach for Speaker-Independent Acoustic Inversion”. In: *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*. 2017, pp. 984–988. url: `http://www.isca-speech.org/archive/Interspeech_2017/abstracts/0804.html`.

[25] Rosanna Turrisi, Raffaele Tavarone, and Leonardo Badino. “Improving generalization of vocal tract feature reconstruction: from augmented acoustic inversion to articulatory feature reconstruction without articulatory data”. In: *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE. 2018, pp. 159–166.

[26] James J. Jiang. “A Literature Survey on Domain Adaptation of Statistical Classifiers”. In: 2007.

[27] Wouter M. Kouw and Marco Loog. “A review of single-source unsupervised domain adaptation”. In: *CoRR* abs/1901.05335 (2019). arXiv: `1901.05335`. url: `http://arxiv.org/abs/1901.05335`.

[28] Sinno Jialin Pan and Qiang Yang. "A survey on transfer learning". In: *IEEE Transactions on knowledge and data engineering* 22.10 (2009), pp. 1345–1359.

[29] Woodland P. C. "Speaker adaptation for continuous density HMMs: A review". In: *ISCA Tutorial and Research Workshop (ITRW) on Adaptation Methods for Speech Recognition*. 2001.

[30] Leggetter C. J. and Woodland P. C. "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models". In: *Computer speech & language* 9.2 (1995), pp. 171–185.

[31] Saon G. et al. "Speaker adaptation of neural network acoustic models using i-vectors". In: *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE. 2013, pp. 55–59.

[32] Gaspard Monge. "Mémoire sur la théorie des déblais et de remblais". In: *Histoire de l'Académie Royale des Sciences de Paris, avec les Mémoires de Mathématique et de Physique pour la même année*. 1781.

[33] Rosanna Turrisi et al. "Multi-source Domain Adaptation via Weighted Joint Distributions Optimal Transport". In: *arXiv preprint arXiv:2006.12938* (2020).

[34] Joel M Gould et al. *Method of speech command recognition with dynamic assignment of probabilities according to the state of the controlled applications*. US Patent 5,960,394. Sept. 1999.

[35] Igor Szoke et al. "Comparison of keyword spotting approaches for informal continuous speech". In: *Ninth European conference on speech communication and technology*. 2005.

[36] Joseph Keshet, David Grangier, and Samy Bengio. "Discriminative keyword spotting". In: *Speech Communication* 51.4 (2009), pp. 317–329.

[37] Mitchel Weintraub. "LVCSR log-likelihood ratio scoring for keyword spotting". In: *1995 International Conference on Acoustics, Speech, and Signal Processing*. Vol. 1. IEEE. 1995, pp. 297–300.

[38] Laura J Ball, Susan Fager, and Melanie Fried-Oken. "Augmentative and alternative communication for people with progressive neuromuscular disease". In: *Physical Medicine and Rehabilitation Clinics* 23.3 (2012), pp. 689–699.

[39] David R Beukelman and Pat Mirenda. *Augmentative & alternative communication: Supporting children and adults with complex communication needs*. Paul H. Brookes Publishing, 2013.

[40] D Jeffery Higginbotham et al. "Access to AAC: Present, past, and future". In: *Augmentative and alternative communication* 23.3 (2007), pp. 243–257.

[41] Howard C Shane et al. "Using AAC technology to access the world". In: *Assistive technology* 24.1 (2012), pp. 3–13.

[42] Cecilia Di Nardi et al. "An automatic speech recognition Android app for ALS patients". In: ().

[43] Lena Maier-Hein et al. “Session independent non-audible speech recognition using surface electromyography”. In: *IEEE Workshop on Automatic Speech Recognition and Understanding, 2005.* IEEE. 2005, pp. 331–336.

[44] Bruce Denby et al. “Prospects for a silent speech interface using ultrasound imaging”. In: *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*. Vol. 1. IEEE. 2006, pp. I–I.

[45] C. P. Browman and L. M. Goldstein. *Towards an articulatory phonology.* 1986, pp. 219–252.

[46] C. P. Browman and L. Goldstein. “Articulatory Phonology: An Overview”. In: *Phonetica* (1992), pp. 155–180.

[47] H. Nam et al. “A procedure for estimating gestural scores from speech acoustics”. In: *The Journal of the Acoustical Society of America* 132.6 (2012), pp. 3980–3989. doi: `10.1121/1.4763545`. eprint: `https://doi.org/10.1121/1.4763545`. url: `https://doi.org/10.1121/1.4763545`.

[48] W. Wang et al. “Unsupervised learning of acoustic features via deep canonical correlation analysis”. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2015), pp. 4590–4594.

[49] L. Badino et al. “Integrating articulatory data in deep neural network-based acoustic modeling”. In: *Computer Speech and Language* 36 (2016), pp. 173–195.

[50] Z. H. Ling, K. Richmond, and J. Yamagishi. “Articulatory control of hmm-based parametric speech synthesis using featurespace-switched multiple regression”. In: *IEEE Transactions on Audio, Speech and Language Processing* 21 (2013), pp. 207–219.

[51] T. Hueber et al. “Speaker adaptation of an acoustic-to-articulatory inversion model using cascaded Gaussian mixture regressions”. In: *14th Annual Conference of the International Speech Communication Association (Interspeech 2013)*. Lyon, France, Aug. 2013, pp. 2753–2757. url: `https://hal.archives-ouvertes.fr/hal-00851894`.

[52] A. Ben-Youssef, H. Shimodaira, and D. A. Braude. “Articulatory features for speech-driven head motion synthesis”. In: Lyon, France, 2013.

[53] Ronald Netsell, Billie Daniel, and Gastone G Celesia. “Acceleration and weakness in parkinsonian dysarthria”. In: *Journal of Speech and Hearing Disorders* 40.2 (1975), pp. 170–178.

[54] NADINE P CONNOR et al. “Parkinsonian deficits in serial multiarticulate movements for speech”. In: *Brain* 112.4 (1989), pp. 997–1009.

[55] Karen Forrest and Gary Weismer. “Dynamic aspects of lower lip movement in Parkinsonian and neurologically normal geriatric speakers’ production of stress”. In: *Journal of Speech, Language, and Hearing Research* 38.2 (1995), pp. 260–272.

[56] Jo Anne Robbins, Jerilyn A Logemann, and Howard S Kirshner. “Swallowing and speech production in Parkinson’s disease”. In: *Annals of Neurology* 19.3 (1986), pp. 283–287.

[57] Kristin K Baker et al. “Thyroarytenoid muscle activity associated with hypophonia in Parkinson disease and aging”. In: *Neurology* 51.6 (1998), pp. 1592–1598.

[58] Vikramjit Mitra et al. “Joint modeling of articulatory and acoustic spaces for continuous speech recognition tasks”. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2017, pp. 5205–5209.

[59] Frank Rudzicz. “Articulatory knowledge in the recognition of dysarthric speech”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.4 (2010), pp. 947–960.

[60] Yılmaz E. et al. “Articulatory features for ASR of pathological speech”. In: *arXiv preprint arXiv:1807.10948* (2018).

[61] Emre Yılmaz et al. “Articulatory and bottleneck features for speaker-independent ASR of dysarthric speech”. In: *Computer Speech & Language* 58 (2019), pp. 319–334.

[62] Weiran Wang, Raman Arora, and Karen Livescu. “Reconstruction of Articulatory Measurements with Smoothed Low-rank Matrix Completion”. In: *IEEE SLT*. Lake Tahoe, Nevada, USA, 2014.

[63] K. Richmond, S. King, and P. Taylor. “Modelling the uncertainty in recovering articulation from acoustics”. In: *Computer Speech and Language* 17.2 (2003), pp. 153–172.

[64] B. Uria et al. “Deep architectures for articulatory inversion”. In: *Proc. of Interspeech*. Portland, Oregon, USA, 2012.

[65] C. Canevari et al. “Cross-corpus and cross-linguistic evaluation of a speaker-dependent DNN-HMM ASR system using EMA data”. In: *Workshop on Speech Production for Automatic Speech Recognition*. Lyon, France, 2013.

[66] Pierre Badin et al. “Visual articulatory feedback for phonetic correction in second language learning”. In: *Second Language Studies: Acquisition, Learning, Education and Technology*. 2010.

[67] X. Xie, X. Liu, and L. Wang. “Deep Neural Network Based Acoustic-to-Articulatory Inversion Using Phone Sequence Information”. In: *Interspeech 2016*. 2016, pp. 1497–1501. doi: `10.21437/Interspeech.2016-659`. url: `http://dx.doi.org/10.21437/Interspeech.2016-659`.

[68] Thomas Hueber et al. “Cross-speaker acoustic-to-articulatory inversion using phone-based trajectory HMM for pronunciation training”. In: *Thirteenth Annual Conference of the International Speech Communication Association*. 2012.

[69] Karen Livescu. “Feature-based pronunciation modeling for automatic speech recognition”. PhD thesis. Massachusetts Institute of Technology, 2005.

[70] C. Y. Liou, J. C. Huang, and W. C. Yang. “Modeling word perception using the Elman network”. In: *Neurocomputing* 71.16 (2008). Advances in Neural Information Processing (ICONIP 2006) / Brazilian Symposium on Neural Networks (SBRN 2006), pp. 3150–3157.

[71] C. Y. Liou et al. “Autoencoder for words”. In: *Neurocomputing* 139 (2014), pp. 84–96. issn: 0925-2312. doi: https://doi.org/10.1016/j.neucom.2013.09.055.

[72] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[73] D. P Kingma and M. Welling. “Auto-Encoding Variational Bayes”. In: (Dec. 2013).

[74] D. P. Kingma and J. L. Ba. “Adam: A Method for Stochastic Optimization”. In: *CoRR* abs/1412.6980 (2014). arXiv: 1412.6980. url: http://arxiv.org/abs/1412.6980.

[75] X. Glorot and Y. Bengio. “Understanding the difficulty of training deep feedforward neural networks”. In: *JMLR W&CP: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*. Vol. 9. Chia Laguna Resort, Sardinia, Italy, May 2010, pp. 249–256.

[76] Victor Zue, Stephanie Seneff, and James Glass. “Speech database development at MIT: TIMIT and beyond”. In: *Speech communication* 9.4 (1990), pp. 351–356.

[77] Douglas B Paul and Janet Baker. “The design for the Wall Street Journal-based CSR corpus”. In: *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*. 1992.

[78] Daniel Povey et al. “The Kaldi Speech Recognition Toolkit”. In: *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Catalog No.: CFP11SRW-USB. Hilton Waikoloa Village, Big Island, Hawaii, US: IEEE Signal Processing Society, Dec. 2011.

[79] Alex Graves and Jürgen Schmidhuber. “Framewise phoneme classification with bidirectional LSTM and other neural network architectures”. In: *Neural networks* 18.5-6 (2005), pp. 602–610.

[80] Hongwei Zhang et al. “Comparison on Neural Network based acoustic model in Mongolian speech recognition”. In: *2016 International Conference on Asian Language Processing (IALP)*. IEEE. 2016, pp. 1–5.

[81] Jeri A Logemann et al. “Frequency and cooccurrence of vocal tract dysfunctions in the speech of a large sample of Parkinson patients”. In: *Journal of Speech and hearing Disorders* 43.1 (1978), pp. 47–57.

[82] Hermann Ackermann and Wolfram Ziegler. “Articulatory deficits in parkinsonian dysarthria: an acoustic analysis.” In: *Journal of Neurology, Neurosurgery & Psychiatry* 54.12 (1991), pp. 1093–1098.

[83] Barbara Bernhardt et al. “Ultrasound in speech therapy with adolescents and adults”. In: *Clinical Linguistics & Phonetics* 19.6-7 (2005), pp. 605–617.

[84] May B Bernhardt et al. “Ultrasound as visual feedback in speech habilitation: Exploring consultative use in rural British Columbia, Canada”. In: *Clinical Linguistics & Phonetics* 22.2 (2008), pp. 149–162.

[85] Nicolas Courty et al. “Joint distribution optimal transportation for domain adaptation”. In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 3730–3739. url: `http://papers.nips.cc/paper/6963-joint-distribution-optimal-transportation-for-domain-adaptation.pdf`.

[86] Yaroslav Ganin et al. “Domain-adversarial training of neural networks”. In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 2096–2030.

[87] Muhammad Ghifary et al. “Deep Reconstruction-Classification Networks for Unsupervised Domain Adaptation”. In: *CoRR* abs/1607.03516 (2016).

[88] Eric Tzeng et al. “Simultaneous Deep Transfer Across Domains and Tasks”. In: *CoRR* abs/1510.02192 (2015). arXiv: `1510.02192`. url: `http://arxiv.org/abs/1510.02192`.

[89] Eric Tzeng et al. “Adversarial Discriminative Domain Adaptation”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[90] Mingsheng Long, Jianmin Wang, and Michael I. Jordan. “Unsupervised Domain Adaptation with Residual Transfer Networks”. In: *CoRR* abs/1602.04433 (2016). arXiv: `1602.04433`. url: `http://arxiv.org/abs/1602.04433`.

[91] Eric Tzeng et al. “Deep Domain Confusion: Maximizing for Domain Invariance”. In: *CoRR* abs/1412.3474 (2014). arXiv: `1412.3474`. url: `http://arxiv.org/abs/1412.3474`.

[92] Nicolas Courty et al. “Optimal Transport for Domain Adaptation”. In: *CoRR* abs/1507.00504 (2015). arXiv: `1507.00504`. url: `http://arxiv.org/abs/1507.00504`.

[93] Bharath Bhushan Damodaran et al. “DeepJDOT: Deep Joint Distribution Optimal Transport for Unsupervised Domain Adaptation”. In: *ECCV 2018 - 15th European Conference on Computer Vision*. Vol. 11208. LNCS. European Conference on Computer Vision 2018 (ECCV-2018). Munich, Germany: Springer, Sept. 2018, pp. 467–483. doi: `10.1007/978-3-030-01225-0\_28`. url: `https://hal.inria.fr/hal-01956356`.

[94] Ievgen Redko, Amaury Habrard, and Marc Sebban. “Theoretical Analysis of Domain Adaptation with Optimal Transport”. In: *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18-22, 2017, Proceedings, Part II*. Ed. by Michelangelo Ceci et al. Vol. 10535. Lecture Notes in Computer Science. Springer, 2017, pp. 737–753. doi: `10.1007/978-3-319-71246-8\_45`. url: `https://doi.org/10.1007/978-3-319-71246-8%5C_45`.

[95] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. “Domain adaptation with multiple sources”. In: *Advances in neural information processing systems*. 2009, pp. 1041–1048.

[96] Judy Hoffman, Mehryar Mohri, and Ningshan Zhang. “Algorithms and theory for multiple-source adaptation”. In: *Advances in Neural Information Processing Systems*. 2018, pp. 8246–8256.

[97] Xingchao Peng et al. “Moment Matching for Multi-Source Domain Adaptation”. In: *arXiv preprint arXiv:1812.01754* (2018).

[98] Han Zhao et al. “Adversarial Multiple Source Domain Adaptation”. In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio et al. Curran Associates, Inc., 2018, pp. 8559–8570. url: `http://papers.nips.cc/paper/8075-adversarial-multiple-source-domain-adaptation.pdf`.

[99] Junfeng Wen, Russell Greiner, and Dale Schuurmans. “Domain Aggregation Networks for Multi-Source Domain Adaptation”. In: *ArXiv* abs/1909.05352 (2019).

[100] Ruijia Xu et al. “Deep Cocktail Network: Multi-source Unsupervised Domain Adaptation with Category Shift”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018). doi: `10.1109/cvpr.2018.00417`. url: `http://dx.doi.org/10.1109/CVPR.2018.00417`.

[101] Chuang Lin et al. “Multi-source Domain Adaptation for Visual Sentiment Classification”. In: *ArXiv* abs/2001.03886 (2020).

[102] Han Zhao et al. “On Learning Invariant Representations for Domain Adaptation”. In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. Long Beach, California, USA: PMLR, 2019, pp. 7523–7532. url: `http://proceedings.mlr.press/v97/zhao19a.html`.

[103] Jinyu Li et al. “An overview of noise-robust automatic speech recognition”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22.4 (2014), pp. 745–777.

[104] Joao Neto et al. “Speaker-adaptation for hybrid HMM-ANN continuous speech recognition system”. In: (1995).

[105] Roberto Gemello et al. “Linear hidden transformations for adaptation of hybrid ANN/HMM models”. In: *Speech Communication* 49.10-11 (2007), pp. 827–835.

[106] Jian Xue, Jinyu Li, and Yifan Gong. “Restructuring of deep neural network acoustic models with singular value decomposition.” In: *Interspeech*. 2013, pp. 2365–2369.

[107] Shaofei Xue et al. “Fast adaptation of deep neural network based on discriminant codes for speech recognition”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22.12 (2014), pp. 1713–1725.

[108] Yong Zhao, Jinyu Li, and Yifan Gong. “Low-rank plus diagonal adaptation for deep neural networks”. In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2016, pp. 5005–5009.

[109] Ossama Abdel-Hamid and Hui Jiang. “Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code”. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. 2013, pp. 7942–7946.

[110] Sree Hari Krishnan Parthasarathi et al. "fMLLR based feature-space speaker adaptation of DNN acoustic models". In: *Sixteenth annual conference of the international speech communication association*. 2015.

[111] Tara N Sainath et al. "Optimization techniques to improve training speed of deep neural networks for large speech tasks". In: *IEEE Transactions on Audio, Speech, and Language Processing* 21.11 (2013), pp. 2267–2276.

[112] Sining Sun et al. "An unsupervised deep domain adaptation approach for robust speech recognition". In: *Neurocomputing* 257 (2017), pp. 79–87.

[113] Z. Meng et al. "Adversarial Teacher-Student Learning for Unsupervised Domain Adaptation". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018, pp. 5949–5953. doi: `10.1109/ICASSP.2018.8461682`.

[114] Zhong Meng, Jinyu Li, and Yifan Gong. "Adversarial speaker adaptation". In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 5721–5725.

[115] Dong Yu et al. "KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. 2013, pp. 7893–7897.

[116] Zhen Huang et al. "Maximum a posteriori adaptation of network parameters in deep models". In: *arXiv preprint arXiv:1503.02108* (2015).

[117] Zhen Huang et al. "Rapid adaptation for deep neural networks through multitask learning". In: *Sixteenth Annual Conference of the International Speech Communication Association*. 2015.

[118] Yann Brenier. "Polar factorization and monotone rearrangement of vector-valued functions". In: *Communications on pure and applied mathematics* 44.4 (1991), pp. 375–417.

[119] L. V. Kantorovich. "On the translocation of masses". In: *Journal of Mathematical Sciences*. 2006. doi: `10.1007/s10958-006-0049-2`.

[120] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein gan". In: *arXiv preprint arXiv:1701.07875* (2017).

[121] Aude Genevay, Gabriel Peyré, and Marco Cuturi. "Learning generative models with sinkhorn divergences". In: *arXiv preprint arXiv:1706.00292* (2017).

[122] Jian Shen et al. "Wasserstein distance guided representation learning for domain adaptation". In: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

[123] Cédric Villani. *Topics in optimal transportation*. 58. American Mathematical Soc., 2003.

[124] Cédric Villani. *Optimal transport: old and new*. Vol. 338. Springer Science & Business Media, 2008.

[125] Gabriel Peyré, Marco Cuturi, et al. “Computational Optimal Transport: With Applications to Data Science”. In: *Foundations and Trends® in Machine Learning* 11.5-6 (2019), pp. 355–607.

[126] Baochen Sun and Kate Saenko. “Deep coral: Correlation alignment for deep domain adaptation”. In: *European conference on computer vision*. Springer. 2016, pp. 443–450.

[127] Shai Ben-David et al. “Impossibility theorems for domain adaptation”. In: *International Conference on Artificial Intelligence and Statistics*. 2010, pp. 129–136.

[128] Yifan Wu et al. “Domain Adaptation with Asymmetrically-Relaxed Distribution Alignment”. In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. Long Beach, California, USA: PMLR, 2019, pp. 6872–6881.

[129] Michael Held, Philip Wolfe, and Harlan P Crowder. “Validation of subgradient optimization”. In: *Mathematical programming* 6.1 (1974), pp. 62–88.

[130] Kilian Fatras et al. “Learning with minibatch Wasserstein: asymptotic and gradient properties”. In: *arXiv preprint arXiv:1910.04091* (2019).

[131] Ievgen Redko et al. “Optimal transport for multi-source domain adaptation under target shift”. In: *International Conference on Artificial Intelligence and Statistics (AISTAT)*. 2019.

[132] Massahi Sugiyama, Matthias Krauledat, and Klaus-Robert M ”uller. “Covariate Shift Adaptation my Importance Weighted Cross Validation”. In: *J. Mach. Learn. Res.* 8 (Dec. 2007), pp. 985–1005.

[133] Adam Paszke et al. “Automatic differentiation in pytorch”. In: (2017).

[134] Rich Caruana. “Multitask Learning”. In: *Machine Learning* 28.1 (1997), pp. 41–75. issn: 1573-0565. doi: `10.1023/A:1007379606734`. url: `https://doi.org/10.1023/A:1007379606734`.

[135] Tamás Grósz et al. “Assessing the degree of nativeness and Parkinson’s condition using Gaussian processes and deep rectifier neural networks”. In: (2015).

[136] J Stone, A Carson, and M Sharpe. “Functional symptoms and signs in neurology: assessment and diagnosis”. In: *Journal of Neurology, Neurosurgery & Psychiatry* 76.suppl 1 (2005), pp. i2–i12.

[137] James R Williamson et al. “Segment-dependent dynamics in predicting Parkinson’s disease”. In: *Sixteenth annual conference of the international speech communication association*. 2015.

[138] Guozhen An et al. “Automatic recognition of unified Parkinson’s disease rating from speech with acoustic, i-vector and phonotactic features”. In: *Sixteenth Annual Conference of the International Speech Communication Association*. 2015.

[139] Ming Tu, Visar Berisha, and Julie Liss. "Interpretable Objective Assessment of Dysarthric Speech Based on Deep Neural Networks." In: *INTERSPEECH*. 2017, pp. 1849–1853.

[140] Milos Cernak et al. "Characterisation of voice quality of Parkinson's disease using differential phonological posterior features". In: *Computer Speech & Language* 46 (2017), pp. 196–208.

[141] Juan Camilo Vásquez-Correa, Juan Rafael Orozco-Arroyave, and Elmar Nöth. "Convolutional Neural Network to Model Articulation Impairments in Patients with Parkinson's Disease." In: *INTERSPEECH*. 2017, pp. 314–318.

[142] KL Kadi et al. "Discriminative prosodic features to assess the dysarthria severity levels". In: *Proceedings of the World Congress on Engineering*. Vol. 3. 2013.

[143] S-A Selouani et al. "Using speech rhythm knowledge to improve dysarthric speech recognition". In: *International Journal of Speech Technology* 15.1 (2012), pp. 57–64.

[144] Abner Hernandez et al. "Dysarthria Detection and Severity Assessment using Rhythm-Based Metrics". In: *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH), Shanghai, China*. 2020, pp. 25–29.

[145] Dirk Van Compernolle. "Noise adaptation in a hidden Markov model speech recognition system". In: *Computer Speech & Language* 3.2 (1989), pp. 151–167.

[146] HM Cung and Yves Normandin. "Noise adaptation algorithms for robust speech recognition". In: *Speech Communication* 12.3 (1993), pp. 267–276.

[147] Sadaoki Furui et al. *Noise adaptation system of speech model, noise adaptation method, and noise adaptation program for speech recognition*. US Patent 7,424,426. 2008.

[148] James Robert, Marc Webbie, et al. *Pydub*. 2018. url: http://pydub.com/.

[149] Brian McFee et al. "librosa: Audio and Music Signal Analysis in Python". In: *Proceedings of the 14th Python in Science Conference*. Ed. by Kathryn Huff and James Bergstra. 2015, pp. 18–24. doi: 10.25080/Majora-7b98e3ed-003.

[150] Ava-Lee Kotler and Nancy Thomas-Stonell. "Effects of speech training on the accuracy of speech recognition for an individual with a speech impairment". In: *Augmentative and Alternative Communication* 13.2 (1997), pp. 71–80.

[151] Nancy J Manasse, Karen Hux, and Joan L Rankin-Erickson. "Speech recognition training for enhancing written language generation by a traumatic brain injury survivor". In: *Brain Injury* 14.11 (2000), pp. 1015–1034.

[152] Parimala Raghavendra, Elisabet Rosengren, and Sheri Hunnicutt. "An investigation of different degrees of dysarthric speech as input to speaker-adaptive and speaker-dependent recognition systems". In: *Augmentative and Alternative Communication* 17.4 (2001), pp. 265–275.

[153] Phil Green et al. “Automatic speech recognition with sparse training data for dysarthric speakers”. In: *Eighth European Conference on Speech Communication and Technology*. 2003.

[154] Cheryl Goodenough-Trepagnier and Michael I Rosen. “Towards a method for computer interface design using speech recognition”. In: *REPORT NO PUB DATE NOTE AVAILABLE FROM* (1991), p. 341.

[155] Mumtaz Begum Mustafa et al. “Severity-based adaptation with limited data for ASR to aid dysarthric speakers”. In: *PloS one* 9.1 (2014), e86285.

[156] Joel Shor et al. “Personalizing ASR for dysarthric and accented speech with limited data”. In: *arXiv preprint arXiv:1907.13511* (2019).

[157] Santiago Omar Caballero Morales and Stephen J Cox. “Modelling errors in automatic speech recognition for dysarthric speakers”. In: *EURASIP Journal on Advances in Signal Processing* 2009.1 (2009), p. 308340.

[158] François Bolley, Arnaud Guillin, and Cédric Villani. “Quantitative concentration inequalities for empirical measures on non-compact spaces”. In: *Probability Theory and Related Fields* 137.3-4 (2007), pp. 541–593.

[159] François Hernandez et al. “TED-LIUM 3: twice as much data and corpus repartition for experiments on speaker adaptation”. In: *International Conference on Speech and Computer*. Springer. 2018, pp. 198–208.

[160] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. “Voxceleb2: Deep speaker recognition”. In: *arXiv preprint arXiv:1806.05622* (2018).

[161] Sonja Körner et al. “Speech therapy and communication device: impact on quality of life and mood in patients with amyotrophic lateral sclerosis”. In: *Amyotrophic Lateral Sclerosis and Frontotemporal Degeneration* 14.1 (2013), pp. 20–25.

[162] Joseph R Duffy. *Motor Speech Disorders E-Book: Substrates, Differential Diagnosis, and Management*. Elsevier Health Sciences, 2019.

[163] Mauro Nicolao et al. “A framework for collecting realistic recordings of dysarthric speech-the homeservice corpus”. In: *Proceedings of LREC 2016*. European Language Resources Association. 2016.

[164] Xavier Menendez-Pidal et al. “The Nemours database of dysarthric speech”. In: *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP’96*. Vol. 3. IEEE. 1996, pp. 1962–1965.

[165] Heejin Kim et al. “Dysarthric speech database for universal access research”. In: *Ninth Annual Conference of the International Speech Communication Association*. 2008.

[166] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. “Sequence to sequence learning with neural networks”. In: *Advances in neural information processing systems*. 2014, pp. 3104–3112.

[167] Lena Hartelius and Per Svensson. “Speech and swallowing symptoms associated with Parkinson’s disease and multiple sclerosis: a survey”. In: *Folia phoniatrica et logopaedica* 46.1 (1994), pp. 9–17.

[168] Aileen K Ho et al. “Speech impairment in a large sample of patients with Parkinson’s disease”. In: *Behavioural neurology* 11.3 (1998), pp. 131–137.

[169] Frederic L Darley, Joe R Brown, and Norman P Goldstein. “Dysarthria in multiple sclerosis”. In: *Journal of Speech and Hearing research* 15.2 (1972), pp. 229–245.

[170] Lena Hartelius, Björn Runmarker, and Oluf Andersen. “Prevalence and characteristics of dysarthria in a multiple-sclerosis incidence cohort: relation to neurological data”. In: *Folia phoniatrica et logopaedica* 52.4 (2000), pp. 160–177.

[171] Bryan J Traynor et al. “Clinical features of amyotrophic lateral sclerosis according to the El Escorial and Airlie House diagnostic criteria: A population-based study”. In: *Archives of neurology* 57.8 (2000), pp. 1171–1176.

[172] Anton Chen and C Gaelyn Garrett. “Otolaryngologic presentations of amyotrophic lateral sclerosis”. In: *Otolaryngology—Head and Neck Surgery* 132.3 (2005), pp. 500–504.

[173] Andressa da Costa Franceschini and Lucia Figueiredo Mourao. “Dysarthria and dysphagia in amyotrophic lateral sclerosis with spinal onset: a study of quality of life related to swallowing”. In: *NeuroRehabilitation* 36.1 (2015), pp. 127–134.

[174] Julien Bogousslavsky, Guy Van Melle, and Franco Regli. “The Lausanne Stroke Registry: analysis of 1,000 consecutive patients with first stroke.” In: *Stroke* 19.9 (1988), pp. 1083–1092.

[175] Robert Teasell et al. “Clinical characteristics of patients with brainstem strokes admitted to a rehabilitation unit”. In: *Archives of physical medicine and rehabilitation* 83.7 (2002), pp. 1013–1016.

[176] Heather L Flowers et al. “The incidence, co-occurrence, and predictors of dysphagia, dysarthria, and aphasia after first-ever acute ischemic stroke”. In: *Journal of communication disorders* 46.3 (2013), pp. 238–248.

[177] Mansi Pankaj Jani and Geeta Bharat Gore. “Occurrence of communication and swallowing problems in neurological disorders: analysis of forty patients”. In: *NeuroRehabilitation* 35.4 (2014), pp. 719–727.

[178] Martha Taylor Sarno. “The nature of verbal impairment after closed head injury.” In: *Journal of Nervous and Mental Disease* (1980).

[179] MT Sarno, A Buonaguro, and E Levita. “Characteristics of verbal impairment in closed head injured patients.” In: *Archives of Physical Medicine and Rehabilitation* 67.6 (1986), p. 400.

[180] Kathryn M Yorkston et al. “The relationship between speech and swallowing disorders in head-injured patients.” In: *The Journal of Head Trauma Rehabilitation* (1989).

[181] Ismail Safaz et al. “Medical complications, physical function and communication skills in patients with traumatic brain injury: a single centre 5-year experience”. In: *Brain Injury* 22.10 (2008), pp. 733–739.

[182] Claire Mitchell et al. "Interventions for dysarthria due to stroke and other adult-acquired, non-progressive brain injury". In: *Cochrane Database of Systematic Reviews* 1 (2017).

[183] Karen Livescu, Preethi Jyothi, and Eric Fosler-Lussier. "Articulatory feature-based pronunciation modeling". In: *Computer Speech & Language* 36 (2016), pp. 212–232.

[184] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory". In: *Neural Comput.* 9.8 (Nov. 1997), pp. 1735–1780. issn: 0899-7667. doi: `10.1162/neco.1997.9.8.1735`. url: `http://dx.doi.org/10.1162/neco.1997.9.8.1735`.