TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies

Adaptation of speech recognition systems to selected real-world deployment conditions

Habilitation thesis

Author: **Ing. Petr Červa, Ph.D.**
Field: Technical cybernetics

Liberec 2021

Declaration

I hereby certify, I, myself, have written my habilitation thesis as an original and primary work using the literature listed below.

I acknowledge that the Technical University of Liberec will make my habilitation thesis public in accordance with Section 47b of Act No. 111/1998 Coll., on Higher Education Institutions and on Amendment to Other Acts (the Higher Education Act), as amended.

I am aware of the consequences which may under the Higher Education Act result from a breach of this declaration.

4. 6. 2021 Ing. Petr Červa, Ph.D.

Adaptace systémů rozpoznávání řeči na vybrané reálné podmínky nasazení

Abstrakt

Tato habilitační práce se zabývá problematikou adaptace systémů rozpoznávání řeči na vybrané reálné podmínky nasazení. Je koncipována jako sborník celkem dvanácti článků, které se touto problematikou zabývají. Jde o publikace, jejichž jsem hlavním autorem nebo spoluatorem, a které vznikly v rámci několika navazujících výzkumných projektů. Na řešení těchto projektů jsem se podílel jak v roli člena výzkumného týmu, tak i v roli řešitele nebo spoluřešitele.

Publikace zařazené do tohoto sborníku lze rozdělit podle tématu do tří hlavních skupin. Jejich společným jmenovatelem je snaha přizpůsobit daný rozpoznávací systém novým podmínkám či konkrétnímu faktoru, který významným způsobem ovlivňuje jeho funkci či přesnost.

První skupina článků se zabývá úlohou neřízené adaptace na mluvčího, kdy systém přizpůsobuje svoje parametry specifickým hlasovým charakteristikám dané mluvící osoby. Druhá část práce se pak věnuje problematice identifikace neřečových událostí na vstupu do systému a související úloze rozpoznávání řeči s hlukem (a zejména hudbou) na pozadí. Konečně třetí část práce se zabývá přístupy, které umožňují přepis audio signálu obsahujícího promluvy ve více než v jednom jazyce. Jde o metody adaptace existujícího rozpoznávacího systému na nový jazyk a metody identifikace jazyka z audio signálu.

Obě zmíněné identifikační úlohy jsou přitom vyšetřovány zejména v náročném a méně probádaném režimu zpracování po jednotlivých rámcích vstupního signálu, který je jako jediný vhodný pro on-line nasazení, např. pro streamovaná data.

Klíčová slova: automatické rozpoznávání řeči, on-line zpracování streamovaných dat, adaptace na mluvčího, detekce řeč/neřeč, rozpoznávání řeči s hudbou na pozadí, identifikace jazyka

Adaptation of speech recognition systems to selected real-world deployment conditions

Abstract

This habilitation thesis deals with adaptation of automatic speech recognition (ASR) systems to selected real-world deployment conditions. It is presented in the form of a collection of twelve articles dealing with this task; I am the main author or a co-author of these articles. They were published during my work on several consecutive research projects. I have participated in the solution of them as a member of the research team as well as the investigator or a co-investigator.

These articles can be divided into three main groups according to their topics. They have in common the effort to adapt a particular ASR system to a specific factor or deployment condition that affects its function or accuracy.

The first group of articles is focused on an unsupervised speaker adaptation task, where the ASR system adapts its parameters to the specific voice characteristics of one particular speaker. The second part deals with a) methods allowing the system to identify non-speech events on the input, and b) the related task of recognition of speech with non-speech events, particularly music, in the background. Finally, the third part is devoted to the methods that allow the transcription of an audio signal containing multilingual utterances. It includes a) approaches for adapting the existing recognition system to a new language and b) methods for identification of the language from the audio signal.

The two mentioned identification tasks are in particular investigated under the demanding and less explored frame-wise scenario, which is the only one suitable for processing of on-line data streams.

Keywords: automatic speech recognition, on-line processing of streamed data, speaker adaptation, speech/non-speech detection, language identification

Acknowledgements

I would like to thank all my colleagues from the Institute of Information Technology and Electronics at the Technical University of Liberec, who have participated in creation of the articles utilized in this thesis, in particular to Jindřich Žďánský, Lukáš Matějů and Jiří Málek.

My biggest thanks go to my colleague Jan Nouza, who was the supervisor of my master and doctoral studies. I have learned from him a lot of my scientific experience as well as life lessons.

Finally, I express my deep and sincere gratitude to my family for their continuous and unparalleled love, help and support.

Contents

List of abbreviations

AM	acoustic model
ASR	automatic speech recognition
BTN	bottleneck
CAE	convolutional autoencoders
DNN	deep neural network
ER	error rate
FAE	fully connected autoencoders
FER	frame error rate
FSMN	feed-forward sequential memory network
G2P	grapheme-to-phoneme
GAČR	Grant agency of the Czech Republic
GD	gender dependent
GMM	Gaussian mixture model
HMM	hidden Markov model
ICASSP	International Conference on Acoustics, Speech and Signal Processing
JMCT	joint multi-condition training of the AM and CAE
LID	language identification
LM	language model
MCT-CAM	multi-condition training of convolutional acoustic models
NAKI	national and cultural identity
RTF	real-time factor
SA	speaker adaptation
SAD	speech activity detection
SAT	speaker adaptive training
SI	speaker independent
SNR	signal-to-noise ratio
TAČR	Technology agency of the Czech Republic
TDNN	time-delay neural network
TSD	text, speech and dialogue
TUL	Technical university of Liberec
WER	word error rate
WFST	weighted finite state transducer

1 Introduction

Automatic speech recognition (ASR) systems are currently the subject of intensive scientific research in the field of machine learning. Depending on the requirements of the target application, they can process the input data in a batch or in a frame-wise mode. In the former case, there are no strict requirements for the speed of transcription. The recognition can thus be performed off-line and even within several recognition passes. This type of processing is, for example, used in systems for transcription and/or indexing of large audio archives. The latter method of recognition is the only possibility for processing of streamed data. The critical factor here is that the ASR system must be able to process the data in real time and often also with just a small delay (latency). Frame-wise applications include, for example, voice aids for physically disabled persons, dictation software or a platform for 24/7 monitoring of TV/R broadcasts.

The choice of speech recognition systems made in the previous paragraph is not random. On the contrary, these are examples of real and genuinely deployed applications. Their basis, in terms of speech recognition, was created within the Laboratory of computer speech processing at TUL. I have also been involved in the development of all these types of systems within our team during my entire career so far. I have been focusing mainly on methods or modules that allow these systems to cope with various real-world operating conditions. These conditions include, e.g., speakers with different voice characteristics, non-speech event or multilingual input data, and negatively affect the accuracy of these systems.

This habilitation thesis takes the form of a collection of 12 selected articles dealing with the above-introduced topic. Their common factor is the effort to adapt the given recognition system to a selected aspect of its real deployment. I participated in the creation of these articles as the main author or co-author in the period from 2011 to 2021. Three of the articles have been published in scientific journals (Radioengineering, Speech Communication and Computer Speech and Language). Eight other papers appeared in proceedings of two major conferences, Interspeech and ICASSP, and the last article was published in proceedings of the TSD conference. The main author or co-author of several of these conference papers is my former PhD student Lukáš Matějů, who has successfully completed his PhD studies recently (in 2020). Another of the co-authors is my current PhD student, František Kynych, who is now in the first year of his studies.

1.1 Main parts of this work

All articles included in this thesis are divided into three main groups according to their topics. Each of them represents one task, in the solution of which I gradually participated in various consecutive research projects.

The first task deals with adaptation of a speech recognition system to a specific speaker. It is referred to as speaker adaptation (SA) and it was the subject of my post-doctoral GAČR project from 2011 to 2013. The outputs of this research have found practical application in the above-mentioned systems for disabled persons, in systems for dictation into a computer (2010–2021), in the system for transcription and indexing of the Czech Radio archive (developed within the NAKI project, 2011–2014), and also in the system for transcription of audiovisual recordings of lectures (TAČR project LeTran, 2011–2014). Speaker adaptation methods have, in all these applications, helped to increase the recognition accuracy for speakers whose voice characteristics differ from those of the speakers in the training database of the ASR system. In addition, voice aids for disabled persons are often used by people with non-standard pronunciation, for whom the adaptation ability of the system is even more important.

The second part of this thesis is devoted to adaptation to non-speech events. I solved this task as a member of the research team within the project TAČR Multilimedia from 2015 to 2017. Since 2018, I have been dealing with it as a co-investigator of the follow-up TAČR project DeepSpot, 2018–2021. One of the common topics for both these projects is the frame-wise (on-line) transcription of TV/R streams. This type of data often contains a large amount of non-speech segments, such as music, or speech segments distorted by music and other noises in the background.

The solution of this problem can be performed in two-steps. First, it is necessary to completely filter out the non-speech segments by using a speech activity detection (SAD) module before the transcription process is started. The use of the SAD module as a continuous filter on the input of an ASR system then not only prevents meaningless transcription of non-speech parts, but also reduces the overall computational demands. The reason is that speech transcription is, by an order of magnitude, computationally more demanding than speech detection. However, it is clear that after SAD-based filtering, the audio signal can still contain some parts with a high level of noise in the background. One of the reasons is that speech utterances in TV/R shows are sometimes deliberately tinted with music. This thesis thus also contains articles devoted to the problem of the transcription of speech with music in the background.

The third group of articles deals with approaches that allow ASR systems to cope with the language variability of input data. For example, when transcribing various Czech and Slovak TV/R programs, the given program or audio stream often contains utterances in both these languages. This is especially true for programs such as interviews or talk shows, where the guest of the show and the presenter do not have to speak the same language. A similar phenomenon of mixing related languages does not occur only in the Czech Republic and Slovakia. It also appears in the states of the former Yugoslavia or the Soviet Union and, for example, in the

Scandinavian countries. The main Scandinavian languages, i.e., Swedish, Norwegian and Danish, are – at least to some extent – mutually intelligible, and it often happens that, for example, a Swedish-speaking person may appear in a Norwegian television program, etc. This issue is therefore valid in the DeepSpot project (focused on Slavic languages) as well as the international NordTrans project, of which I am also a co-investigator and which in particular focuses on transcription of speech in Norwegian and Swedish.

The articles devoted to language variability, that are included in this thesis, address two sub-tasks. Only the solution of both of them allows the ASR system to process data containing utterances in multiple languages. The first sub-task concerns building an ASR system for a new language as efficiently as possible, i.e., by adapting an ASR system that already exists for a similar language or a group of similar languages. The second sub-task deals with language identification (LID) from acoustic data. Two different operating modes are considered here: batch as well as more difficult and unexplored frame-wise (on-line) processing. Based on the results of this detection, proper language-specific components of the ASR system can then be employed for speech recognition.

1.2 Other related works and articles not included in this thesis

In addition to the three tasks mentioned so far, I have also worked on other problems related to the topic of this habilitation thesis in recent years. Papers dealing with these additional topics often have a practically oriented content and I decided not to include them in this thesis. Yet, they are worth a brief mention.

One of such adaptation tasks is a proper transcription of new and, at the same time, semantically important words that constantly appear in the media. These words typically include the names of newly medialized persons (namely politicians), emerging companies, geographic locations, or professional terms related to breaking events, etc. As a part of the DeepSpot project, we proposed a working solution of this problem, described in [1]. It consists in fully automating the scheme of daily updates of the lexicon and subsequently adapting the language model of the ASR system. I implemented this approach into a software tool, which includes a number of partial processes and operations, and which has been practically deployed for several languages and several years alongside the entire platform for monitoring of TV/R broadcasts.

Another publication [2] dealt with an exactly opposite problem. As part of the NAKI project, which focused on the automatic transcription and indexing of the Czech Radio archive since 1920, it was necessary to adapt the recognition system (especially its dictionary and language model) towards the language of the past historical periods.

2 Adaptation of ASR systems to selected deployment conditions

2.1 Basic components of ASR systems

In general, ASR systems [3] consume an audio signal on the input and produce a corresponding sequence of text labels on the output. The input signal is processed by a parameterization module, which performs its segmentation into frames and then calculates a vector of features for every frame. The output labels may represent, e.g., letters or words (the latter case is considered in this thesis) from the lexicon L of the system.

Formally, this problem can be formulated as

$$\hat{W} = \underset{W \in L^*}{\operatorname{argmax}}\, p(W|X) \tag{2.1}$$

where $\hat{W} = \{w_1, ..., w_N\}$ is the output sequence of N words, $X = \{x_1, ..., x_T\}$ is the sequence of input feature vectors of length T, $p(W|X)$ is the posterior probability of W given X, and L^* is the set of all possible sequences that can be constructed from the words in the lexicon L.

There exist two main ASR system architectures that allow to estimate $\hat{W}$ according to 2.1. The systems that directly estimates the posterior probability by mapping X into $\hat{w}$ are called as end-to-end [4, 5]. They are usually represented by a selected type of deep neural network (DNN).

The second type of ASR systems that are included in the subject of this work utilize Bayes' theorem. In this case, eq. 2.1 can be rewritten as

$$\begin{aligned} \hat{W} &= \underset{W \in L^*}{\operatorname{argmax}} \frac{p(X|W)p(W)}{p(X)} \\ &= \underset{W \in L^*}{\operatorname{argmax}}\, p(X|W)p(W) \end{aligned} \tag{2.2}$$

where $p(X)$ is the marginal probability of observing X, $p(W)$ is the prior probability of W, and $p(X|W)$ is the likelihood of X given W.

In many cases, this type of architecture utilizes hidden Markov models (HMMs) under the hood. The HMM-based architecture comprises three independent parts: the acoustic, pronunciation and language models. All of them are language dependent. Their brief descriptions are as follows:

The acoustic model (AM) provides mapping between feature vectors and the corresponding sequence of phonemes. The pronunciation model is used to construct a lexicon, which maps phonemes to individual words from the lexicon. Finally, the language model (LM) represents probability distribution over sequences of words.

In fact, the HMM mechanism itself is utilized only within AM. In this case, eq. 2.2 can be rewritten as

$$\begin{aligned} \hat{W} &= \underset{W \in L^*}{\operatorname{argmax}}\, p(X|W)p(W) \\ &= \underset{W \in L^*}{\operatorname{argmax}}\, p(X, W) \\ &= \underset{W \in L^*}{\operatorname{argmax}}\, p(X, S, W) \end{aligned} \tag{2.3}$$

where $S = \{S_t \in \{1, .., J\} | t = 1, ..., T\}$ is the HMM state sequence and J is the number of states of the HMM model. Eq. 2.3 can be further expressed as

$$\begin{aligned} \hat{W} &= \underset{W \in L^*}{\operatorname{argmax}} \sum_S p(X, S, W) \\ &= \underset{W \in L^*}{\operatorname{argmax}} \sum_S p(X|S, W)p(S|W)p(W)) \end{aligned} \tag{2.4}$$

By using conditional independent hypothesis, $p(X|S, W)$ can be approximated as $p(X|S, W) \approx p(X|S)$. After that, eq 2.4 becomes

$$\hat{W} = \underset{W \in L^*}{\operatorname{argmax}} \sum_S p(X|S)p(S|W)p(W)) \tag{2.5}$$

where the term $p(X|S)$ represents the above-mentioned acoustic model, the $p(S|W)$ stands for the pronunciation model (which is encoded in the lexicon), and $p(W)$ corresponds to the language model.

The HMM model assumes observation independence (i.e., observations at any time only depend on the hidden state at that time). By using this fact and the probability chain rule, $P(X|S)$ can be decomposed as:

$$\begin{aligned} p(X|S) &= \prod_T p(x_t|x_1, .., x_{t-1}, S) \\ &\approx \prod_T p(x_t|s_t) \end{aligned} \tag{2.6}$$

Similar to $p(W|X)$, there exist two different AM architectures to calculate the likelihood $p(x_t|s_t)$. The first utilizes Gaussian mixture models (GMMs). The resulting topology is then called GMM-HMM. Since 2012–2013, the state-of-the-art architectures have been using DNNs. The resulting DNN-HMM topology [6] calculates $p(x_t|s_t)$ by using the posterior probability $p(s_t|x_t)$ and prior probability $p(s_t)$ according to

$$p(x_t|s_t) \propto \frac{p(s_t|x_t)}{p(s_t)} \tag{2.7}$$

The ASR systems described so far must comprise one more important module, the decoder. This is the core component that decodes the optimal $\hat{W}$ over all possible solutions L^* given all the three models.

Most of today's decoders allow for frame-wise processing, i.e., they are able to operate in real-time and over streams of frames. This capability diminishes the differences between on-line processing of streamed data and off-line processing of recordings because the off-line data can also be streamed to the input of the frame-wise system. The only difference is that an off-line recording can be split into N parts, which may then be transcribed separately. The results can then be available in the $N - times$ shorter time.

2.2 Additional modules allowing practical deployment

As mentioned in the first chapter, each deployed ASR system must cope with a large variability of the input data, which may decrease its recognition accuracy. One possible solution to this problem is to complement the ASR system with additional components. This approach is illustrated in Fig. 2.1, where these components (described in red) comprise the SA module, the LID module, the SAD module (the speech/non-speech detector) and the training module employing the adaptation techniques. Each of these modules helps the ASR system to suppress the selected source or sources of the input data variability.

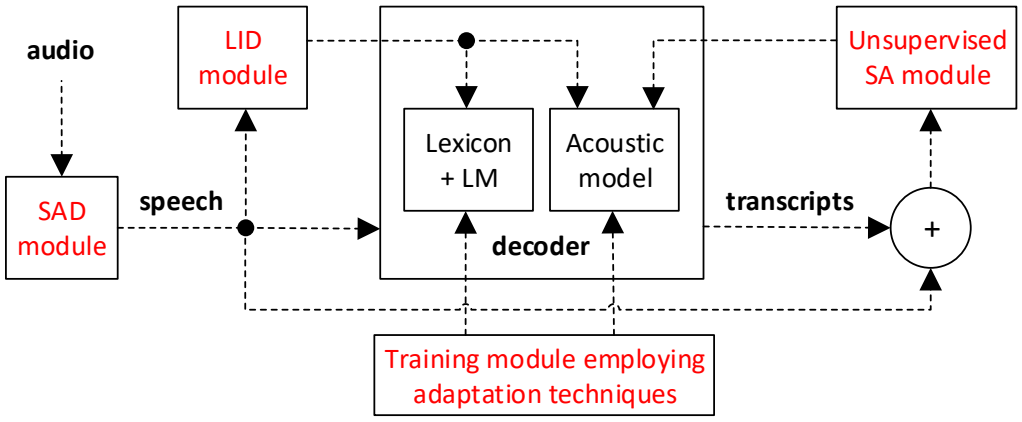


Figure 2.1: Principal scheme of practical deployment of an ASR system. The core decoder module is complemented by additional modules (described in red), which are the subject of this thesis and, at the same time, allow the ASR system to cope with different sources of the input data variability. Of course, the real use of each of the modules depends on the nature of the target application.

2.2.1 Unsupervised speaker adaptation module

The unsupervised SA module is connected on the output of the ASR system and consumes the sequences of recognized words. In fact, it is a feedback connection, because the output of this module updates the parameters of the AM that is used for recognition. We can see that this module utilizes not only the recognized sequence of words, but also the input speech data.

The depicted adaptation scheme is called unsupervised, because the transcriptions of the speech data are not checked by a human annotator. The unsupervised SA task can be expressed as

$$\hat{\theta} = \underset{\theta \in \theta^*}{\operatorname{argmax}}\, p(\theta|X, \hat{W}) \tag{2.8}$$

where $\hat{\theta}$ represents the resulting set of parameters of the acoustic model that are adapted to the particular speaker, θ^* is the set of all possible values of these parameters and $p(\theta|X, \hat{W})$ is the posterior probability of θ given the decoded sequence $\hat{W}$ and the feature sequence X; the latter represents the speech belonging to the particular speaker only.

From the practical point-of-view, the unsupervised SA mode is more complicated than the supervised one. The reason is that the unsupervised SA module has to cope with various additional problems, among which the most important one is that $\hat{W}$ may contain wrong words. In a system transcribing multi-speaker audio data, such as TV/R programs, another issue is ensuring the main assumption in 2.8, i.e., that SA must be performed on a single-speaker basis only.

From the processing point of view, unsupervised SA can be performed on-line (during transcription) as well as off-line (before transcription). The first article included in this thesis proposes an incremental SA approach suitable in the former mode. The target application is a single speaker system for transcription of lectures. The remaining three articles deal with the more conventional off-line task. However, the target ASR system processes multi-speaker recordings in this case. The proposed adaptation scheme thus incorporates a speaker diarization module, which forms the input to the SA module. It segments the transcribed data into speaker homogeneous parts and performs clustering of these single-speaker segments in order to increase the amount of available adaptation data for each speaker.

2.2.2 Speech/non speech detection module

The SAD modules considered in this thesis operate on the frame-wise basis only. The reasons are twofold. First, this way of processing corresponds to the nature of the task so that it is also suitable for batch (off-line) mode. Second, it is the only suitable option for live processing of broadcast streams.

A frame-wise SAD module works as a filter – it consumes the input feature vectors frame-by-frame and passes out only the feature vectors containing speech. The speech frames are then utilized in all subsequent modules, i.e., in the LID module, in the decoder and also in the SA module – see Fig. 2.1. That means

that the accuracy of the SAD module is a critical factor, because it influences the performance of all other components.

From the mathematical point of view, the frame-wise speech/non speech detection may be formulated as a binary classification problem

$$\hat{c}_t = \underset{c_t \in \{speech, non-speech\}}{\operatorname{argmax}} p(c_t | x_{t-m}, ..., x_t, ...x_{t+m}) \tag{2.9}$$

where $\hat{c}_t$ is the output label at time t and $p(c_t|x_{t-m}, ..., x_t, ...x_{t+m})$ is the posterior probability of c_t given the partial sequence of feature vectors $x_{t-m}, ..., x_t, ...x_{t+m}$ of length $M = 2m + 1$. The SAD approaches included in this thesis (see Sect. 4.1) employ discriminative DNN-based models to estimate this posterior probability directly, i.e., without the use of Bayes' theorem. The parameter M then represents the context's size that the DNN utilizes over its input. The higher the value of m is, the higher latency the SAD module has. In other words, the SAD module a non-causal system is for $m > 0$ – the output label is determined using m feature vectors following the current frame at time t.

However, practical experience shows that the frame-wise SAD module can not make the decision on the level of individual feature vectors alone. In other words, the resulting sequence $\hat{C}$ of labels for all the input vectors can not simply be created by concatenating individual labels c_t over time. The reason follows from the fact that the label at each time t depends not only on the values of several surrounding frames, but also on the several previous labels.

When the wrong assumption of unlimited time independence of labels is used, the classifier tends to oscillate with a high frequency between the two classes for some non-stationary parts of the input data. At the same time, it may output correct labels for some other, longer parts containing stationary signal (such as long lasting speech segments with a low level of the background noise). Under such scenarios, the overall accuracy achieved on the frame level may be high, but, unfortunately, the SAD module may not be practically usable as it is not possible to send an oscillating output signal to the subsequent modules.

Therefore, the performance of the SAD module must also be evaluated in terms of the change point detection. Moreover, a confusion matrix must be used in this case because a proper SAD module must achieve a high value of the recall as well as a high value of the precision.

In practice, all these facts lead to a solution when the output from the classifier operating over individual frames must be smoothed. The solution we adopt for this purpose relies on a decoder based on a weighted finite state transducer (WFST). The role of the WFST-based decoder is then similar to the decoder in an ASR system: it produces an optimal output sequence of labels from all possible sequences.

The task may then be reformulated as:

$$\hat{C} = \underset{C \in C^*}{\operatorname{argmax}}\, p(C|X) \tag{2.10}$$

where $\hat{C}$ is a decoded sequence of speech/non-speech labels, C^* is the set of all possible sequences of labels and X is the sequence of all input contexts of width

M. The total latency of the SAD module is then given by m, usually 0.5s, plus the latency of the decoder, usually around 1.5s.

2.2.3 Language identification module

Spoken language identification is the task of correctly determining the language spoken in a speech utterance [7]. The goal of the LID module in Fig 2.1 is thus to provide the decoder with information of which language-specific components should be used for speech decoding.

The LID module can be utilized for batch processing and well as within frame-wise processing of streamed data. The articles included in this thesis only solve both of these tasks from the acoustic modeling point of view. We do not utilize any phonotactics information.

The frame-wise LID task is unexplored in the literature. However, it can be formulated in a way similar to frame-wise SAD. The main difference is that the number of output classes may be higher than two and that these classes are much more difficult to distinguish (particularly for languages that are related to each other, such as Slavic languages).

Therefore, our solution of this task takes advantage of similar components as the scheme we use for frame-wise SAD. It utilizes a binary or multi-class DNN-based language classier whose output is smoothed by a WFST-based decoder (more details are provided in Sec. 5.2. The frame-wise LID module then must also be evaluated in terms of language-change point detection accuracy.

In a batch processing (off-line) mode, it is supposed that all the input data is monolingual. In other words, the LID module outputs only one language label. This label can also easily be determined by using a DNN-based frame-wise classifier. In this case, its output is not smoothed by any decoder. The global label is simply calculated by averaging the probability vectors obtained for each frame of the input data. The language with the maximum average probability then corresponds to the output label.

2.2.4 Training module employing adaptation techniques

All of the previous three modules operate during the deployment phase of the given ASR system. On the contrary, the training module utilizes adaptation techniques within the training period, which always precedes practical deployment. As mentioned above in Sec.1.2, an example of the adaptation employed in the training phase is the process of daily updates of the lexicon and of the language model. However, the articles included in this thesis employ the adaptation in training for three additional main purposes.

Speaker-adaptive training

The first purpose is the so-called speaker-adaptive training (SAT). The goal of this technique is in general to optimize a set of speaker dependent parameters together

with a set of speaker independent (SI) parameters in order to remove speaker variation. In the deployment phase, the model created in SAT yields, after adaptation to a particular speaker, a better level of performance than the model created by adapting the conventional speaker independent model.

The SAT scheme can be performed using various SA methods. However, from the points of view regarding practical and computations demands, it is advantageous to use, e.g., constrained maximum likelihood linear regression over feature vectors (fMLLR) [8]. The reason is that, in this case, it is necessary to compute and store just one transformation matrix for each training speaker.

In the first step of the fMLLR-based SAT, the fMLLR transformation is estimated for each speaker occurring in the training database. After that, the set of speaker specific transformations is used for accumulating training statistics over all available training data. In this phase, the training data is split into single-speaker chunks that are processed after applying the corresponding speaker-specific transformations. Finally, the collected training statistics are utilized for estimation of the new SI model with reduced speaker variation.

Multi-condition training

The second purpose is the adaptation of the AM for recognition of distorted speech. In the training phase, this type of adaptation can be performed by multi-condition training. In this framework, all considered distorted speech signals are included in the training set, i.e., the model incorporates knowledge on all of the possible interference types. Considering environmental noise, this multi-condition training was reported to obtain high performance in [9]. Besides additive noise, this technique was also demonstrated to be beneficial for reverberated speech in [10]. Sec.4.2 includes two articles dealing with utilization of multi-condition training (and some other techniques) for recognition of speech with background music. However, we also used this framework for robust recognition of conversational telephone speech in [11].

Adaptation to a new language

The third purpose is adaptation of an existing modular ASR system to a new language. In this scenario, we need to create the above-mentioned language-dependent modules: a pronunciation model (encoded in a lexicon), a language model, and an acoustic model. Preparing the first two components is the simpler part of the work since we can utilize large amounts of texts that are publicly available on the Internet. The best sources are web-pages of major newspapers and broadcasters. For most languages, it is not difficult to collect a corpus with several gigabytes of texts, make a list of words occurring in them, create a lexicon from the most frequent ones, and compute the corresponding LM. For the pronunciation model, we need to learn the basic rules that describe relationships between orthographic and phonetic forms of words and compile a grapheme-to-phoneme (G2P) transducer. The rule-based (canonical) pronunciations usually work well in the initial development phase and later they can be augmented by variants that occur in real speech recordings.

Preparing the AM is a more complicated task. A good AM needs to be trained on spoken data that cover thousands of speakers, various topics, different acoustic conditions and at least several tens (or better, hundreds) of hours of speech. Moreover, all recordings must be annotated on the acoustic-phonetic level, i.e., as sequences of phonemes and noises. This annotation can be done automatically if we know the orthographic transcriptions. It is possible to purchase speech databases suitable for AM training for many languages (e.g. Globalphone [12]), but they are often too small, relatively expensive or contain utterances from environments that do not correspond to the target application. Hiring hundreds of native speakers that would record a large amount of speech is an alternative, but its cost is also high.

Therefore, Sec. 5.1 in this thesis describes an approach we have developed in order to minimize the above-described issues and costs by automating the process of collection, phonetic transcription and AM training for new languages. Out method employs AMs existing for other (related) languages and takes advantage of the public Internet sources containing real talks recorded in authentic environments.

3 Unsupervised speaker adaptation task

This part of the thesis includes four articles dealing with unsupervised speaker adaptation methods. All of them were published within my postdoctoral GAČR project (2011–2013), which was focused on speaker adaptation methods in speech recognition systems.

As mentioned in Sec. 2.1, the architecture of AMs employed by the state-of-the-art speech recognition systems has changed since that time. The main difference is that the generative GMM topology has been replaced by discriminative models based on DNNs [13]. These models are significantly more robust and also more difficult to adapt (they belong to the class of discriminative models). Nevertheless, some of the techniques proposed or investigated within the four articles, such as the previously mentioned SAT, can also be employed within DNN training [14, 15].

3.1 On-line incremental unsupervised SA

In the first Interspeech paper [16], an unsupervised incremental speaker adaptation approach is proposed. This method is evaluated within a single-speaker system for lecture transcription. The aim is to adapt the AM of this system for each input lecture during its transcription (the process of adaptation on 10 minutes of data takes only a few seconds).

The proposed adaptation process runs as follows: The ASR module stores the processed feature vectors and phonetic forms of recognized words. When the amount of collected data is increased by a defined value (e.g., each ten minutes), a two-phase adaptation procedure is started. In its first-phase, the last updated model, all available feature vectors, and the corresponding phonetic transcriptions are employed to calculate the accumulators needed for estimating fMLLR. After that, the updated fMLLR matrix and feature vectors with the corresponding phonetic transcripts are employed to accumulate statistics for model-based adaptation of the mean vectors. The resulting model with adapted parameters and fMLLR matrix is then employed for decoding and in the subre-sequent iteration of the adaptation.

The SI model is trained discriminatively using 300 hours of speech recordings belonging to 3,940 male and 2,030 female speakers. The features are 13-dimensional MFCCs with Δ and $\Delta\Delta$. A square HLDA matrix is employed for decorrelation of the feature space. The used sampling frequency is 16 kHz.

The method is evaluated on a test set compiled from 17 hours of lecture recordings. Their average duration is 1.4 hours. By using incremental adaptation with a

step of 10 minutes during each lecture, the word error rate (WER) of the baseline speaker independent (SI) system is significantly reduced from 29.8% to 22.5%.

In addition to incremental SA, the ASR system optimized for lecture transcription also utilizes domain specific languages models. These LMs are created for lectures on two topics, economics and informatics, by linear interpolation of several sub-models to minimize the perplexity of the mixed model on a development set. They yield a WER value that is absolutely by 1% lower than the general LM. Note that the mentioned sub-models are estimated from various corpora including theses, newspaper articles, transcriptions of lectures, web discussions, etc.

3.2 Unsupervised SA for off-line transcription

The next three contributions are focused on unsupervised speaker adaptation for off-line transcription.

Two-step unsupervised feature-based SA

The first Interspeech paper [17] deals with unsupervised feature-based speaker adaptation techniques. The investigated methods include vocal tract length normalization (VTLN) [18, 19] and fMLLR. Both of these methods are evaluated separately and combined together not only in the speech recognition process, but also for building the normalized acoustic model within the SAT scheme.

The resulting approach is performed in two-steps: in the first one, VTLN and the corresponding normalized SI model (created within VTLN-based SAT) are employed for speech transcriptions of each input utterance. The resulting phonetic transcription is then employed in the second step, which relies on estimation of the global fMLLR transform. The second recognition pass is then performed using this transformation and the SI model created within the fMLLR-based SAT.

The SI acoustic model used is based on the tied-state context dependent HMMs of Czech phonemes and several types of noises. It contains 3,062 physical states with up to 32 components per state – the total number of Gaussians is approximately 98k. The feature vector is composed of 39 MFCC parameters (13 static coefficients and their first and second derivatives). The sampling frequency is 16 kHz. The training speech database is compiled from 120 hours of speech recordings belonging to 1,128 male and 755 female speakers.

Experimental evaluation is performed on several types of TV/R recordings, e.g. broadcast news, talk-shows, parliament speeches, etc. Their total length is 10.9 hours and they contain 94,321 words. All these recordings are manually segmented into 3,083 parts belonging mostly to a single speaker. Their average length is 12.7 seconds.

The obtained results show that the two-step approach yields an additional decrease in WER over VTLN as well as over fMLLR. While VTLN reduces WER of the baseline SI system from 17.8% to 16.6 % and CMLLR to 15.2 % , the two-pass adaptation combining VTLN and fMLLR decreases WER to 14.7 %. This value corresponds to a relative reduction over the baseline SI model by 17.6 %.

Note that the next advantage of the proposed SA method is that it allows for more accurate pruning of hypotheses during the speech recognition. The real-time factor of the ASR system is 1.13 for the baseline SI system, 1.03 for the system employing VTLN and just 0.97 when VTLN is followed by fMLLR.

Incorporation of the speaker diarization module

The second paper, published also in Interspeech proceedings [20], studies close incorporation of automatic speaker diarization with unsupervised SA approaches. It provides the motivation for utilization of speech transcripts in the diarization process and analyzes the effect it yields in terms of diarization performance or computational cost. For a better insight, the limit performance is evaluated by substituting most of the components of the automatic system by the oracle ones.

The resulting diarization approach is proposed to be performed in several consecutive steps. The first one is the speech transcription. In this phase, we make use of the information about word boundaries and also take advantage of classification of various non-speech events as provided by the ASR system. This allows the diarization system to neglect various noises produced by speakers (breathing, various hesitation sounds, cough, lip-smack, etc.) that carry no speaker-specific information and thus interfere with the representation of clusters. Within the second step, speaker change-points are detected. In this case, the information about the start position of transcript elements is used as the constraining condition for the choice of change-point candidates. After speaker segmentation, the process of speaker clustering is carried out using the two-stage method described in [21].

Experimental results presented in [20] show that the utilization of speech transcripts in the diarization process yields improvement in both segmentation and clustering performance. From the SA point of view, the performance achieved using the proposed diarization techniques was close to the performance achieved by systems using the oracle components.

Speaker-adaptive speech recognition for large spoken archives

Finally, given all the findings from the previous two works, a new speaker-adaptive speech recognition scheme is proposed in article [22], which was published in the Speech Communication Journal. In this paper, the scheme is developed within a series of consecutive experiments. The resulting final form is depicted in Fig. 3.2.

The method is based on integration of automatic speaker diarization and adaptation methods and is designed to achieve a low real-time factor of the entire adaptation process. It thus employs just two decoding passes, where the first one is carried out using the lexicon with a reduced number of items (to the level of 20,000 words). Moreover, the transcripts from the first pass serve not only for adaptation, but also as the input to the above-described two-stage speaker diarization module. The output of diarization is then utilized for a cluster-based unsupervised SA approach, which can be described as follows:

At first, given the transcripts from the first SI decoding pass, the global fMLLR is employed to estimate the transform for every speaker detected by the diarization

module. In this step, the appropriate gender dependent (GD) model is used as initial for fMLLR instead of the SI model. Both GD models and speaker-specific transforms are then used in the second decoding pass, where each transform is applied to all feature vectors belonging to the corresponding speaker. When the amount of the adaptation data assigned to a given speaker is below the threshold required for the estimation of the transformation matrix, only the appropriate GD model is used for the decoding. Moreover, the GD models are created within the SAT scheme to further improve the recognition accuracy.

The architecture of the GD models is based on the tied-state context dependent HMMs of Czech phonemes and several types of non-speech events. This model contains 4k physical states with up to 32 Gaussian components per state (i.e., 120k components in total). It is trained using 300 h of speech recordings sampled at 16 kHz. The features are again 13-dimensional MFCCs with Δ and $\Delta\Delta$.

The experimental evaluation of the proposed methods is performed on 26 TV/R recordings of daily news, talk shows, political debates and regional news reports. Their total length amounts to 846 minutes. On this data-set, the resulting SA method yields a reduction in WER from 22.24% to 18.85% over the SI system.

Finally, it should also be noted that the whole unsupervised SA scheme was deployed in practice within our project NAKI, 2011–2014, whose goal was to transcribe and made public for browsing 100,000 hours of recordings covering 90 years of broadcasting of the Czech Radio.

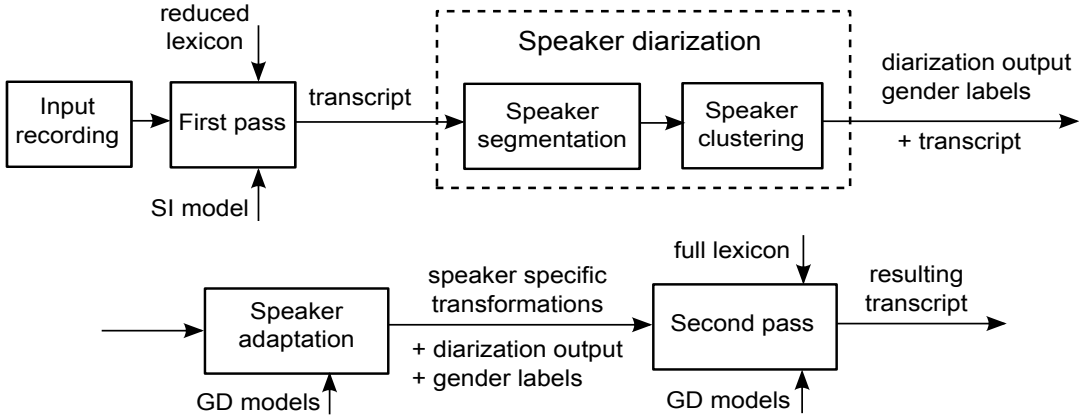


Figure 3.1: The scheme of the speaker-adaptive ASR approach for improved transcription of large spoken archives as published in [22].

3.3 Reprints

[16] P. Cerva, J. Silovsky, J. Zdansky, J. Nouza, and J. Malek. "Real-Time Lecture Transcription using ASR for Czech Hearing Impaired or Deaf Students". In: *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012. ISCA*, pp. 763–766.

[17] P. Cerva, K. Palecek, J. Silovsky and J. Nouza. "Using Unsupervised Feature-Based Speaker Adaptation for Improved Transcription of Spoken Archives". In: *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011. ISCA*, pp. 2565–2568.

[20] J. Silovsky, P. Cerva, J. Zdansky, and J. Nouza. "Study on Integration of Speaker Diarization with Speaker Adaptive Speech Recognition for Broadcast Transcription". In: *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012. ISCA*, pp. 478–481.

[22] P. Cerva, J. Silovsky, J. Zdansky, J. Nouza, and L. Seps. "Speaker-adaptive speech recognition using speaker diarization for improved transcription of large spoken archives". In: *Speech Communication 55.10, 2013*, pp. 1033–1046.

ISCA Archive
http://www.isca-speech.org/archive

INTERSPEECH 2012
ISCA's 13th Annual Conference
Portland, OR, USA
September 9-13, 2012

Real-Time Lecture Transcription using ASR for Czech Hearing Impaired or Deaf Students

Petr Cerva, Jan Silovsky, Jindrich Zdansky, Jan Nouza and Jiri Malek

Institute of Information Technology and Electronics, Faculty of Mechatronics,
Technical University of Liberec, Studentska 2, CZ 461 17, Liberec, Czech Republic
{petr.cerva,jan.silovsky,jindrich.zdansky,jiri.malek,jan.nouza}@tul.cz

Abstract

This paper describes a client-server system developed to enable hearing impaired persons to participate in lectures by providing real-time displayed transcripts. The core of this system is formed by an ASR module running on a recognition server and processing the input audio-video stream. This engine utilizes a large lexicon, topic-specific language models mixed properly from various sources (e.g. transcripts of spontaneous utterances, theses, web discussions) and unsupervised incremental speaker adaptation methods to cope with spontaneous lecture speech in highly inflective Czech language. The raw output of the ASR module is converted into a more readable form using a developed post-processing module based on finite state transducers. The resulting formated text (i.e. containing punctuation marks, digit strings, etc) is then displayed on the screen of each client device (e.g. notebook or tablet) in the lecture room.

Index Terms: real-time lecture transcription, applications for handicapped persons, applications in learning

1. Introduction

Automatic transcription and processing of spoken lectures has attracted a lot of attention recently. For example, several complex solutions [1, 2] have been developed to improve the quality and accessibility of higher education by allowing students to browse the content of (academic) lectures over Internet.

These systems utilize ASR technology to create automatic transcription of input lecture recordings. These are usually processed off-line in several recognition passes [3, 4] in order to achieve the best possible recognition accuracy. The created time-aligned transcripts are then indexed and users can search in them using specialized web interfaces from the comfort of their homes. It has also been shown, that on-line video lecture recordings have positive impact on student performance [5].

There is one group of handicapped people, hard hearing impaired or deaf students, for whom the existence of web lecture browsers is very important. Unfortunately, although the previously mentioned systems enable students with hearing difficulties access to the content of lecture recordings, they still do not give them the opportunity to participate in live lectures in university classrooms.

The solution of this problem consists in simultaneous automatic conversion of spoken lectures into text. This text can then be displayed on a large screen at front of the classroom or utilized for creating subtitles that can be streamed together with the given audio-video data to student laptops or tablets.

Significant effort for real-time lecture processing is being conducted within the Liberated Learning Consortium [6] that includes universities and companies from all around the world. Their solution utilizes IBM ViaScribe [7] and its available for U.S. English, Chinese Mandarin, Spanish, and Arabic. Several other languages are under development.

In this paper, we present our system that can be employed for real-time transcription of lectures in inflective Czech language. Its core is formed by an ASR module developed originally for a dictation system that is available on the Czech and Slovak market. We also describe the process of building a language model (LM) that can be utilized by this engine for lectures. After that, we present unsupervised incremental speaker adaptation approach that we employ to improve recognition accuracy without the need of supervised speaker enrollment. This method and the created LMs are then evaluated on 18 hours of Czech lectures on economics a informatics. Finally, we describe our post-processing module.

2. System architecture

The principal scheme of our system is depicted in Fig. 1. The given lecture is captured and streamed over the HTTP protocol using the VLC player[1]. The video and audio is encoded using the H.264[2] and the MP3[3] codec respectively and encapsulated in the MP4 container.

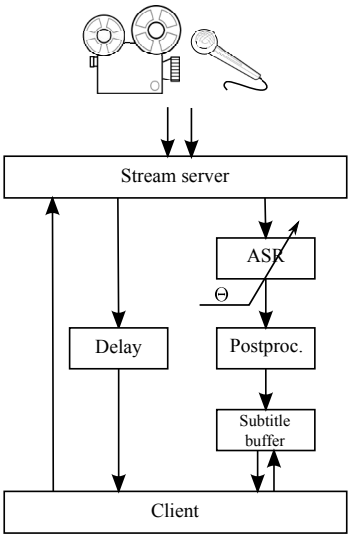


Figure 1: *Scheme of the real-time lecture transcription system.*

[1] www.videolan.org/vlc/
[2] We use the FFmpeg (http://ffmpeg.org/).
[3] We use the Lame encoder (http://lame.sourceforge.net/)

The ASR module processes the data stream and produces subtitles that are post-processed and then stored in a subtitle buffer. That means that subtitles are not muxed into the broadcasted stream but they are provided on demand as driven by the javascript logic at the client side. The symbol Θ in Fig. 1 denotes the recognition pruning threshold that is changed adaptively according to the actual value of the decoding delay to ensure that the ASR module operates in real-time. The total delay caused by streaming, recognition and post-processing is around 5 seconds.

3. Optimization of the ASR module for lectures

When developing a new ASR system for lecture transcription or optimizing an existing one, one has to deal with several difficulties [8]. These include namely high degree of spontaneity in lecture speech or various acoustic conditions in lecture rooms. In the following two subsections, we address some of these issues from the language and acoustic modeling point of view.

3.1. Language modeling

There are several main reasons, why the task of building a proper LM for lectures is very challenging. Firstly, the most lectures contain technical and specialized words that are not common in spontaneous speech while specialized texts lack many words with high occurrence in spontaneous lectures. The next reason is that lecture speech contains a lot of spontaneous effects like filler or partial words and word contractions or reductions. Finally, there is a lack of lecture transcripts or text data that would cover the previously mentioned type of speech. Within this work, we try to solve these difficulties by building LMs that are optimized for one selected target domain (i.e. lectures on economics and informatics in our case).

For this purpose, we acquired two development text corpora. The first one contains verbatim transcripts of 10 lectures on economics (83k words), while the second one contains 6 lectures on informatics (38k words). We also collected a large training text database (see Tab. 1). It includes corpora that a) have at least small degree of spontaneity (e.g. broadcast news or verbatim lecture transcripts on various topics) or that b) contain specialized words and phrases from the target domain (e.g. newspaper articles on economics or informatics). After text preprocessing, we utilize this database not only for training of LMs but also for creation of a lexicon suitable for lectures. All these steps are described below in more detail.

3.1.1. Text preprocessing

The preprocessing module deletes formating tags, normalizes characters (e.g. – to -), expands abbreviations, normalizes words (e.g. *museum* to *muzeum*), replaces digits by their spelling, removes punctuations and converts the text into lower case.

3.1.2. Lexicon selection

Our existing word repository for broadcast news transcription contains 1200k of items including common Czech words, foreign names, shortcuts, etc. It has been created during several recent years. Because Czech is highly inflective language, we had to employ tools for automatic inflection in the creation process. Despite of its size, we found the repository to be insufficient with respect to the task of lectures transcription. Therefore, we had to further extend it.

For this purpose, we found more than 6420k of new distinct tokens in all preprocessed text corpora. Obviously, it was not possible to process all of them manually. Hence, we created a list sorted according to their frequency of occurrence and we processed manually just first several thousands of tokens. After that, we used a) Slovak and English spell-checkers to filter out words in these languages as our text corpora are mostly contaminated by these and b) black lists created in recent years to filter out typical typing errors. In total, we processed manually more than 150k of the most frequent resulting tokens and we added by about 70k of new words or new non-standard pronunciation forms to the initial repository.

Finally, we created the general 504k lexicon for lectures. This is formed by 500k entries from the extended repository (chosen according to their frequency of occurrence in all available corpora) and 4k of Czech most frequent collocations.

3.1.3. Estimation of domain-specific LMs

Firstly, we built temporal LMs for all individual training text corpora (summarized in Tab. 1) using the 504k lexicon. After that, we performed two linear interpolations of these models. The weights for these interpolations were calculated so as to minimize the perplexity of the mixed model on the given development set using the SRILM toolkit [9].

Results obtained using created models are presented in section 4.1 and resulting weights for individual temporal models are summarized in Tab. 1. Values in this table show that verbatim lecture transcripts play very important role in both domain-specific LMs. Also texts from monitoring of broadcast news and thesis seems to be very useful while weights of remaining corpora are usually limited to several percent (namely for the model specialized on economics).

Table 1: *Training text corpora and their weights in domain-specific language models.*

type of the corpus	**size [MB]**	**weight in LM for**	
		informatics	**economics**
thesis	4600	0.21	0.12
web discussions	2400	0.08	0.03
monitoring and transcripts			
broadcast programs	1700	0.12	0.48
articles			
economics	250	-	0.03
computers	210	0.07	-
verbatim transcripts			
telephone talks	3	0.03	0.05
lectures	4	0.45	0.21
radio debates	5	0.04	0.08

3.2. Acoustic modeling

Our ASR system uses an acoustic model based on tied-state context dependent HMMs of Czech phonemes and several types of non-speech events. Within this paper, this model contained 4k of physical states with up to 32 Gaussian components per state (i.e. 120k of components in total). The model was trained discriminatively using 300 hours of speech recordings belonging to 3940 male and 2030 female speakers. The features were 13-dimensional MFCCs with Δ and $\Delta\Delta$. We used square HLDA matrix for decorrelation of the feature space. The used sampling frequency was 16 kHz.

3.2.1. Unsupervised incremental speaker adaptation approach

Our adaptation approach (evaluated in sec. 4.2) is performed for each input lecture as unsupervised and in an incremental mode so that the given lecturer is not asked to read any text and the system adapts its parameters gradually during recognition. We utilize global CMLLR transformation of feature vectors that is followed by model-based adaptation of mean vectors using combination of MLLR and MAP.

The adaptation process runs as follows: The ASR module stores the processed feature vectors (not transformed by CMLLR) and phonetic forms of recognized words. When the amount of collected data increases by a defined value (e.g. each ten minutes), the following two-phase adaptation procedure is started. In the first-phase, the last updated model, all available untransformed feature vectors, corresponding phonetic transcriptions and the Baum-Welch algorithm are employed to calculate accumulators needed for estimation of global CMLLR. After that, the model with updated CMLLR and all untransformed feature vectors with corresponding phonetic transcripts are employed again for the second phase of the Baum-Welch algorithm. The resulting new accumulators then serves for model-based adaptation of mean vectors using MLLR with a binary regression tree that is followed by MAP. The resulting model with updated CMLLR and with adapted parameters is then employed for decoding and in the next iteration of adaptation.

4. Experimental evaluation

For evaluation purposes, we collected several lectures on economics and informatics from e-learning archives of several Czech universities (see Tab. 2). They were recorded in several recent years, in different quality and using clip-on microphone. They are not included in the training text database, were narrated by different lecturers and are on different topics than lectures from the development sets.

Table 2: *Lectures chosen for experimental evaluation*

Domain	# of lectures	# of words	length [hours]
economics	10	81436	13.1
informatics	7	73545	11.1
total	17	154981	24.2

4.1. Evaluation of language models

The first experiment was performed using the general LM for lectures that was estimated for the 504k lexicon using all training corpora from Tab. 1. Then, we used the two domain-specific LMs created by mixing as described in section 3.1.3. All the obtained results in terms of Word Error Rate (WER) and perplexity are summarized in Tab. 3. They show that specialized LMs yielded better results than the general LM.

Table 3: *WER [%] obtained using the general and domain-specific LMs*

Domain	OOV	general LM		specialized LMs	
		WER	perpl.	WER	perpl.
informatics	3.9	36.8	4649	35.7	3061
economics	3.3	25.8	12327	24.5	7648
total	3.6	31.0	-	29.8	-

Unfortunately, although the domain-specific LMs had much lower perplexity than the general model, they yielded WER lower just by 1.2 % absolutely. We believe that this results can further be improved by transcription of other lectures, because they have big weight in domain-specific LMs (see Tab. 1).

4.2. Evaluation of speaker adaptation

The evaluation of incremental SA was performed for each input lecture using the corresponding domain-specific LM. The first experiment (see Tab. 4) shows WER reached after a) using one-phase CMLLR or MLLR+MAP based adaptation and b) two-phase adaptation utilizing combination of all these methods. In all cases, new iteration of adaptation was performed each time, when the amount of transcribed data increased by ten minutes.

Table 4: *Results obtained using various incremental speaker adaptation approaches with the domain-specific LMs*

Adapt. method	WER [%] for		
	economics	inform.	total
One-phase adaptation			
CMLLR	21.3	30.3	25.6
MLLR+MAP	20	29	24.2
Two-phase adaptation			
CMLLR + MLLR + MAP	19.2	26.2	22.5

The results show that CMLLR as well as MLLR+MAP led to significant improvement of WER over the baseline SI system and that the combination of all these methods yielded additional reduction of WER. In total, WER of the SI model declined after SA from 29.8% to 22.5% (by 24% relatively).

Unfortunately, the disadvantage of the two-stage adaptation is its computational complexity as it is necessary to process all available adaptation data twice in each iteration. In the next experiment (see Tab. 5), we thus present results obtained when just several iterations of adaptation were performed and the rest part of each lecture was then transcribed using the model created in the last iteration. The iteration step was again 10 minutes.

Table 5: *WER [%] after various number of iterations using combination of CMLLR with MLLR+MAP*

Domain	# of iterations (used amount of data [min])				
	1 (10)	2 (20)	3 (30)	6 (60)	9-12 (all)
informatics	30.5	29.0	27.9	27.1	26.2
economics	21.4	21.0	20.5	19.7	19.2
total	25.7	24.8	24.0	23.2	22.5

These results show that significant reduction of WER over the SI system is reached when just 10 minutes of data is used for adaptation and that WER then converges slowly to the best possible value with increasing number of iterations.

In our practical implementation, we thus perform adaptation if the amount of data reaches values of 5, 15 and 60 minutes. For further acceleration of the adaptation process, our implementation of the Baum-Welch algorithm utilizes pruning and the accumulated data are processed per blocks with fixed length of 30 seconds. Moreover, adaptation data are split into four parts and processed separately by a second computer that does not perform speech decoding. The two-phase adaptation on 10 minutes of data then takes 15 seconds (measured using 4 cores of a machine with Intel Core i7-2600K@3.4GHz).

5. Text post-processing module

It has been shown that the output from the recognizer is very difficult to read and comprehend for hearing impaired persons [10]. For this reason, we have developed a post-processing module (utilized also in our dictation system) that converts the recognized text into a more readable form containing formated digit strings, units, punctuation, etc.

To represent relations between the recognized text and the desired post-processed output, we use weighted Finite State Transducer (FST) models, namely we utilize the OpenFst library[4]. It allows to build FSTs from probabilistic n-gram models (e.g. for automatic punctuation, uppercasing) as well as from hand-written context dependent rewrite rules (e.g. for formating of numbers) and combine them together using common FST operations like composition, concatenation, etc. For compiling of hand-written grammars into FSTs, we employ the tool OpenGrm Thrax[5]. It supports for assigning sub-grammars into variables that can be then utilized in other grammars using FTS operations to produce the final transduction model.

For example, one can define grammar units converting words like 'meters per second' into 'm/s' and grammar numbers producing digit strings from their read form. Concatenation of numbers with units then creates a new grammar that rewrites defined units only when preceded by numbers.

There is a problem that composition of all desired models/rules together produces large models - typically 1GB+ in the OpenFst format (i.e. 30M of states and 60M arcs). That is why we divided all of them into several layers that are applied sequentially during decoding and have clearly defined purpose (e.g.uppercasing). Moreover, this approach also makes the post-processing module very flexible as the final user can easily turn on/off individual layers without computationally intensive FST composition. The layers used for lectures are summarized with several examples of their input and rewritten output in Tab. 6.

The post-processing is performed using on-the-fly composition of the transduction and the input model of unknown size that is possible since the input text is considered to be a linear-topology, unweighted, epsilon-free acceptor. After each composition step, the shortest-path (considering tropical semiring) determined in the resulting model is compared with all other alternative hypotheses. When a common path is found among these hypotheses (i.e. with the same output), the corresponding concatenated output labels are marked as the final fixed output. Since the rest of the best path is not sure, it is denoted as a temporary output (i.e. it can further be changed). Note that due to the multi-layer architecture, the temporary output from one layer is expected to be the regular input to the next layer.

Table 6: *Post-processing layers and examples of their input and rewritten output (translated into English)*

Layer	Input : Rewritten output
numbers and dates	'thousand three' : '1003' 'third October' : '3. October'
units	'2 kilometers per hour' : '2 km/h'
uppercasing	'river hot' : 'river Hot'
titles	'He is a professor' : 'He is a professor' 'professor Jan' : 'Prof. Jan'
auto-punctuation	'yes but I' : 'yes, but I'

[4]http://www.openfst.org/twiki/bin/view/FST/WebHome
[5]http://www.openfst.org/twiki/bin/view/GRM/Thrax

Therefore the decoder contains support for stepping-back in the composition process. Moreover, advanced garbage-collection of the resulting model was developed to get small and constant memory footprint for very long texts.

6. Conclusions

In this paper, we described our system developed for real-time transcription of Czech spoken lectures. Its ASR and post-processing modules can also be adopted for lecture transcription in other languages. We plan to utilize them at least for similar Slavic languages like Slovak or Polish.

The complete system is now being tested by several target users. The first experiences have shown that the reached WER is suitable but not satisfactory. This motivates us to collect more spontaneous data for acoustic as well as language model training. We have obtained very positive response to the post-processing output. Hence we plan to further improve it. A new layer should provide formatting of simple math terms and equations.

7. Acknowledgements

This work was supported by the Czech Science Foundation (project no. P103/11/P499) and by the Technology Agency of the Czech Republic (project no. TA01011142).

8. References

[1] J. Glass, T. J. Hazen, S. Cyphers, I. Malioutov, D. Huynh, and R. Barzilay, "Recent progress in the MIT spoken lecture processing project," in *Proceedings of InterSpeech 2007*. ISCA, 2007, pp. 2553 – 2556.

[2] I. Szoke, J. Cernocky, M. Fapso, and J. Zizka, "SPEECH@FIT lecture browser," in *Proceedings of the 2010 IEEE Spoken Language Technology Workshop*, ser. IEEE Catalog Number: CFP 10SLT-USB. IEEE Signal Processing Society, 2010, pp. 157–158.

[3] L. Lamel, E. Bilinski, J. L. Gauvain, G. Adda, C. Barras, and X. Zhu, "The LIMSI RT07 lecture transcription system," R. Stiefelhagen, R. Bowers, and J. Fiscus, Eds. Berlin, Heidelberg: Springer-Verlag, 2008, ch. Multimodal Technologies for Perception of Humans, pp. 442–449.

[4] C. Fűgen, M. Wőlfel, J. W. McDonough, S. Ikbal, F. Kraft, K. Laskowski, M. Ostendorf, S. Stűker, and K. Kumatani, "Advances in lecture recognition: The ISL RT-06S evaluation system," in *Proceedings of Interspeech 2006*. ISCA, 2006, pp. 1229–1232.

[5] M. B. Wieling and W. H. A. Hofman, "The impact of online video lecture recordings and automated feedback on student performance," *Comput. Educ.*, vol. 54, pp. 992–998, May 2010.

[6] (2012) The liberated learning consortium website. [Online]. Available: http://liberatedlearning.com

[7] K. Bain, S. Basson, A. Faisman, and D. Kanevsky, "Accessibility, transcription, and access everywhere," *IBM Syst. J.*, vol. 44, no. 3, pp. 589–603, Aug. 2005. [Online]. Available: http://dx.doi.org/10.1147/sj.443.0589

[8] A. Park, T. J. Hazen, and J. R. Glass, "Automatic processing of audio lectures for information retrieval: Vocabulary selection and language modeling," in *Proc. ICASSP*, 2005.

[9] A. Stolcke, "SRILM - An extensible language modeling toolkit," in *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002)*, 2002, pp. 901–904.

[10] M. Wald, P. Boulain, J. Bell, K. Doody, and J. Gerrard, "Correcting automatic speech recognition errors in real time," *International Journal of Speech Technology*, vol. 10, no. 1, pp. 1–15, 2007. [Online]. Available: http://eprints.ecs.soton.ac.uk/12140/

Using Unsupervised Feature-Based Speaker Adaptation for Improved Transcription of Spoken Archives

Petr Cerva, Karel Palecek, Jan Silovsky, Jan Nouza

Institute of Information Technology and Electronics, Faculty of Mechatronics,
Technical University of Liberec, Studentska 2, CZ 461 17, Liberec, Czech Republic
petr.cerva,karel.palecek,jan.silovsky,jan.nouza@tul.cz

Abstract

This paper deals with unsupervised feature-based speaker adaptation techniques. The goal is to design an optimal adaptation approach for improving the recognition accuracy of a LVCSR system developed for automatic transcription of large archives of spoken Czech (e.g. the archive of the parliament talks, historical archives of Czech broadcast stations, etc.) For this purpose, several modifications of VTLN and CMLLR techniques were investigated and combined together. Our study focuses on the application of the adaptation methods in the recognition process as well as in building a normalized acoustic model within the speaker adaptive training scheme. The methods were evaluated experimentally on a large amount of various data (with total number 93k words). The resulting two-step adaptation scheme yields a significant WER reduction from 17.8 % to 14.8 %.

Index Terms: unsupervised speaker adaptation, VTLN, CMLLR, SAT, spoken data transcription

1. Introduction

Large archives of spoken data are one of the challenging application areas for speech processing technology. They require complex solutions where speech recognition is combined with data indexation and information retrieval techniques. The systems of this type have been applied, for example, to enable access to national spoken language archives [1] or for collections of historical testimonies [2].

We have been working on the automatic transcription task since 2005 and some of the developed tools have been already deployed for broadcast data mining [3]. One of our current projects aims at processing the archive of historical and contemporary recordings of the Czech Radio. This archive covers almost 90 years of broadcasting (since 1923) and contains hundreds of thousands spoken documents, from which a large portion should be transcribed.

The standard procedure for automatic transcription of an audio document runs as follows: At first, the signal is parameterized and then segmented into speech and non-speech parts (e.g. long silence or music). The speech segments pass into the module searching for changes in signal character, which can be speaker turns and/or changes in the signal band-with (e.g. telephone talks). The found change points are used for splitting the speech into individual utterances, which are transcribed using the given acoustic and language models.

The aim of this work was to develop an efficient speaker adaptation (SA) approach that could be utilized for improving of the transcription accuracy the above mentioned single-speaker utterances. From the SA point of view, the task is challenging, namely because a) the length of individual utterances varies in the range from a few to tens or even hundreds of seconds, b) there is no a priori information on the speakers occurring in the archive documents, and c) the adaptation has to be performed as unsupervised. On the other hand, usually it is not necessary to perform the transcription in real time. It runs off-line so that adaptation can be accomplished in several passes.

There exist two main groups of techniques that can be employed for this kind of adaptation. The methods belonging to the first group, e.g. MLLR or Speaker Selection Training (SST) [4], are sometimes called as model-based. They utilize the transcription of the input utterance and/or statistics collected during training together with the speaker recognition module to update the model parameters. Within this paper, we have adopted the second group of so called feature-based techniques. These can be used for adapting the feature vectors without changing parameters of the model, which can be very large particularly for triphones. Their next advantage is the possibility of easy application for building the normalized acoustic model within Speaker Adaptive Training (SAT) [5] scheme.

This paper is structured as follows: the next section describes the basic principles of SA methods that were adopted within this paper. The section 3 and 4 deals with their experimental evaluation and the resulting two-step adaptation approach is described and evaluated in the section 5.

2. Review of Used Feature-Based SA Techniques

2.1. VTLN

The aim of the Vocal Tract Length Normalization (VTLN) method [6] is to compensate for different lengths of the vocal tract of individual speakers. This is accomplished by warping the frequency axis during signal parameterization. For MFCC parameters, this warping is applied to the magnitude spectrum before the mel frequency scaling is performed. It has been shown [7], that the piece-wise linear warping function is suitable in most situations. This function is represented by one parameter, the warping factor α, and it is modified at the upper boundary frequency f_0 as the linear warping would lead to some filters being placed outside the frequency range.

In practice, Maximum Likelihood Estimation (MLE) has to be used to find the optimal value of α, because there does not exist any "universal" speaker to which the spectra could be normalized. The resulting equation has the form

$$\hat{\alpha} = \arg\max_{\alpha} P\left(\mathbf{X}|\mathbf{T}, \lambda\right) \quad (1)$$

where $\mathbf{X}^{\alpha}$is the warped speech signal with corresponding word transcription $\mathbf{T}$ and λ is the acoustic model used to estimate $\hat{\alpha}$.

2.1.1. VTLN during training

Better results after VTLN based adaptation can be obtained, when the same kind of warping is also applied during the training procedure. Each warped (normalized) testing utterance $\mathbf{X}^{\hat{\alpha}}$ can then be recognized with a normalized model λ_{SAT}, which is trained on the warped data.

This process of normalized model training belongs to the group of SAT techniques and it is usually performed in one or several iterations [6] as follows:

1. An initial acoustic model $\lambda_{\mathbf{0}}$ is estimated on all available non-normalized data belonging to R training speakers.
2. All these training data are divided into R speaker specific sub-sets. Each sub-set then contains just the data belonging to one training speaker.
3. The warping factor $\hat{\alpha}$ is found for each speaker r using his/her acoustic data $\mathbf{X_r}$ with known transcription $\mathbf{T_r}$ as
$$\hat{\alpha_r} = \arg\max_{\alpha} P\left(\mathbf{X}_r^{\alpha}|\mathbf{T}_r, \lambda_0\right) \quad (2)$$
4. A new normalized acoustic model λ_{SAT} is estimated on all available training acoustic data, which are normalized for each speaker r using the found optimal warping factor $\hat{\alpha_r}$.

2.1.2. Unsupervised VTLN + SAT

During unsupervised adaptation mode, the word transcription of adaptation data is not known. The classical way how to perform adaptation in this situation is to employ a two-pass recognition approach: in the first pass, automatic word transcription $\hat{\mathbf{T}}$ of the data is created using the given speech recognizer and non-normalized speaker independent acoustic model. After that, the warping factor can be estimated as

$$\hat{\alpha} = \arg\max_{\alpha} P\left(\mathbf{X}^{\alpha}|\hat{\mathbf{T}}, \lambda_{SAT}\right) \quad (3)$$

In the second pass, the testing utterance is normalized with the found warping factor $\hat{\alpha}$ and speech recognition is performed with the normalized acoustic model λ_{SAT}.

The main disadvantage of this approach is the required computation time, which is necessary for the two recognition passes. Recently, two main approaches have been proposed to solve this problem. Both of them utilize Gaussian Mixture Models (GMMs).

In the first one, one GMM is estimated during SAT in a similar way as the model λ_{SAT} for speech recognition: at first, all available training acoustic data for each speaker are normalized using the corresponding optimal warping factor $\hat{\alpha_r}$ and then, one normalized GMM $\mathbf{\Omega}_{NORM}$ is created using standard ML training on these warped data. During the recognition phase, each utterance is normalized with all possible values of α and the optimal value $\hat{\alpha}$ is selected as

$$\hat{\alpha} = \arg\max_{\alpha} P\left(\mathbf{X}^{\alpha}|\mathbf{\Omega}_{NORM}\right) \quad (4)$$

In the second case [7], the warping factor is determined for all training speakers and all data with the same warping factor are pooled. One GMM $\mathbf{\Omega}^{\alpha}_{NON-NORM}$ is then trained on the corresponding set of non-normalized acoustic data and it represents the distribution of non-normalized feature vectors of the given warping factor in the acoustic space. During testing, the optimal warping factor $\hat{\alpha}$ has to be selected as

$$\hat{\alpha} = \arg\max_{\alpha} P\left(\mathbf{X}|\mathbf{\Omega}^{\alpha}_{NON-NORM}\right) \quad (5)$$

2.2. CMLLR

The aim of the Constrained Maximum Likelihood Linear Regression (CMLLR) method [8] is to reduce the mismatch between an initial acoustic model and the target speaker.

The transformation is constrained as the transformation matrix applied for adaptation of means has to be the same as the one used for adaptation of variances. Therefore the adaptation can also be performed in the feature space.

The adapted feature vector $\hat{\mathbf{o}}$ can be expressed as

$$\hat{\mathbf{o}} = \mathbf{W}\xi \quad (6)$$

where $\mathbf{W}$ is the extended transformation matrix, $\xi = \begin{bmatrix}\omega & o_1 & o_2 & \dots & o_n\end{bmatrix}^{\top}$ is the extended vector of features, n is the dimension of data and ω represents an offset.

The matrix $\mathbf{W}$ has to be calculated in an iterative process, where the likelihood of adaptation data with known transcription is maximized [8]. In the unsupervised mode, this transcription has to be created using the speech recognizer at first.

The advantage of CMLLR is that it can be applied directly in the feature space. When multiple transformations are used, it is only necessary to include Jacobian of each transformation in the likelihood calculation.

2.2.1. SAT using CMLLR

Another advantage of feature-based transformation is that it can be implemented more easily during SAT than unconstrained MLLR, which updates model parameters.

At first, CMLLR transformation is estimated for each speaker occurring in the training database. After that, these speaker specific transformations and equation (6) are used for calculating training statistics over all training data. Finally, these collected statistics and standard formulae are utilized for estimating the new model parameters in the same way as during standard maximum likelihood training.

3. Experimental Setup

All experiments were performed on several types of spoken archives, e.g. broadcast news prepared within the European COST278 project [9], radio news, talk-shows, parliament speeches, etc. These recordings contained not only the clean speech but also a lot of spontaneous utterances. Their total length was 10.9 hours and they contained 94,321 words. All these recordings were segmented manually into 3083 parts belonging mostly to a single speaker. The length of these segments varied in the range from several to several tens of seconds. The average length was 12.7 seconds.

For recognition, we employed our own ASR system operating with lexicon containing 315,105 items with multiple pronunciations. The language model was based on smoothed bigrams estimated on a corpus compiled from Czech (mainly newspaper) texts.

The used SI acoustic model was based on tied-state context dependent HMMs of Czech phonemes and several types of noises. It contained 3,062 physical states with up to 32 components per state - the total number of Gaussians was approximately 98k. The feature vector was composed of 39 MFCC parameters (13 static coefficients and their first and second derivatives). The sampling frequency was 16 kHz.

The training speech database was compiled from 120 hours of speech recordings belonging to 1128 male and 755 female speakers.

4. Evaluation of Individual Methods

4.1. VTLN in training

We used the piece-wise linear warping function with the upper boundary frequency $f_0 = 7\,\mathrm{kHz}$. The warping factors $\hat{\alpha_r}$ were estimated for every training speaker in one iteration of the approach described in the section 2.1.1. Within this process, a grid search was performed with the step 0.02 in the range from 0.8 to 1.2. We employed the single Gaussian model, because it has been shown [6] that the models with higher number of components per state are almost able to capture all the different warping factors for all training speakers. The resulting histogram over all scale factors for all training speakers is shown in Fig. 1.

Finally, one normalized GMM $\mathbf{\Omega}_{NORM}$ and the set of non-normalized GMMs $\mathbf{\Omega}^{\alpha}_{NON-NORM}$ were trained using the values of $\hat{\alpha_r}$ and normalized/non-normalized acoustic data.

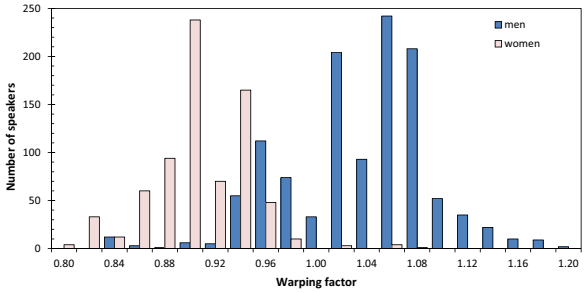


Figure 1: *Warping factors for all training speakers*

4.2. Unsupervised VTLN in testing

The first performed experiment (Tab. 1) compares different approaches for warp factor estimation. The optimal value of the warping factor was calculated for each testing utterance a) according to (3) using the recognized phonetic transcription, b) according to (4) with one normalized GMM and c) according to (5) using the set of non-normalized GMMs.

All GMMs had 128 Gaussian components and the recognition with found $\hat{\alpha}$ was performed with the normalized SI model created in complete retraining procedure on all warped training recordings. The baseline WER of the SI model was 17.8 %.

Table 1: *Comparison of Different Approaches for Warp Factor Estimation during unsupervised VTLN*

Used approach for $\hat{\alpha}$ estimation	**WER [%]**
according to (3) using recognized transcription	16.3
one normalized GMM	16.9
the set of non-normalized GMMs	16.7

We can see from results of this experiment that although the best WER was achieved in the first case, the most suitable approach for practical application is the third one, because the calculation of $\hat{\alpha}$ with the set of non-normalized GMMs is much faster than speech decoding with the baseline SI model.

The aim of the next experiment (see Tab. 2) is to investigate which model should be used for speech recognition with the given determined optimal warping factor. For this purpose, we employed a) the normalized model created in complete retraining procedure on all warped data as in the previous case, b) normalized model created by retraining of the baseline model just in 5 iterations and c) the baseline non-normalized SI model.

Table 2: *Using different types of normalized models for recognition with the determined warping factor*

The type of the model	**WER [%]**
normalized in complete training	16.7
normalized in 5 iterations of retraining	16.6
baseline non-normalized SI	17.5

The results show that the use of the normalized model created in SAT is very important. There was not any significant improvement of results when the baseline non-normalized model was applied. The second result is that it is not necessary to train the normalized model during complete training procedure - it is possible just to retrain the baseline model in a few iterations.

The overall result of both the previous experiments is that unsupervised VTLN and SAT can reduce WER significantly from 17.8 % to 16.6 % - by 7 % relatively.

4.3. Unsupervised CMLLR

At first, CMLLR was evaluated without SAT. The aim was to investigate which kind of CMLLR is the best one for our task, where the amount of adaptation data is limited (12.7 seconds in average for our testing database).

We used the baseline SI model as prior for adaptation and also for transcription of each utterance in the first speech recognition pass. We evaluated CMLLR with two different regression trees, CMLLR utilizing two static classes, CMLLR with one global transform and CMLLR with two different block-diagonal transforms. The results are presented in Tab. 3.

They show that due to the short average length of utterances, the best WER was obtained using only one global transformation. The number of estimated parameters was too high for CMLLR with more than one class and too low for the block-diagonal and diagonal transforms. The adapted model was not estimated properly in all these cases.

Table 3: *Results for unsupervised adaptation using different types of CMLLR*

Used kind of CMLLR	**WER [%]**
8 classes + regression tree	17.1
4 classes + regression tree	16.6
2 static classes (phones / noises + silence)	16.2
one global transform for all units (39x39)	16.0
global block-diagonal (3x13x13 param.)	16.4
global diagonal (39x1x1 param.)	18.2

4.4. Unsupervised CMLLR + SAT

The next experimental evaluation was focused on CMLLR and SAT. The best approach for CMLLR according to previous results (i.e. one global transform) was utilized within training in two different ways. In the first one, CMLLR transformations were estimated for each training speaker using the baseline SI model. These transformations were then utilized for retraining this model in 5 iterations. In the latter case, SAT was performed near in the same way, only CMLLR transformations were updated between every two training iterations.

The obtained results showed only slight differences between the two approaches. The first approach achieved WER of 15.4 %, the latter one 15.2 %. The more important result is that CMLLR based SAT yielded significant additional decrease

in WER over the previous experiments, where the best obtained WER was 16.0 %.

The final WER 15.2 % corresponds to relative reduction over the baseline SI model by 15 %. This is two times better value than the best relative decrease in WER achieved by VTLN (see the section 4.2). On the other hand, CMLLR based adaptation is much more computationally intensive than VTLN - it is necessary to perform one more speech recognition pass.

5. Resulting Two-Step Adaptation Approach

The performed experiments showed that unsupervised VTLN as well as CMLLR reduces WER significantly. Therefore our next idea was to investigate, if it is possible to further improve the results by combining the best two previous approaches.

The resulting two-step approach for recognition utilizes all the previous findings and it is performed as follows: in the first step, VTLN and the corresponding normalized SI model are employed to create the phonetic transcription of the testing utterance. In the second step, global CMLLR transform is estimated. Finally, the second recognition pass is performed using this transform and the model created in CMLLR based SAT.

The described approach was evaluated using several smaller lexicons in the first recognition pass (see Tab. 4). The aim was to reduce the high computation time required for two-pass decoding with the full 315k lexicon. The real time (RT) factor was measured using one core of an Intel Core i7 920 processor.

Table 4: *Evaluation of the two-step adaptation approach.*

the first speech recognition pass with SAT and VTLN						
lexicon's size	30k	40k	50k	100k	150k	**315k**
OOV [%]	7.9	6.1	5.1	2.6	1.8	**1.0**
RT factor	0.56	0.58	0.59	0.76	0.83	**1.03**
WER [%]	27.0	24.1	21.0	18.7	17.7	**16.6**
the second pass with SAT, CMLLR and 315k lexicon						
WER [%]	15.6	15.2	**14.8**	14.7	14.7	**14.7**

The results show that the two-step approach yielded additional decrease in WER over VTLN as well as CMLLR. With the 315k lexicon, VTLN reduced WER to 16.6 %, CMLLR to 15.2 % (see the previous experiment) and two pass adaptation combining VTLN and CMLLR to 14.7 %. This value corresponds to relative reduction over the baseline SI model by 17.6 %. The adaptation also improved pruning. The RT factor was 1.13 for the baseline SI system, 1.03 for VTLN (see Tab. 4) and 0.97 for combination of VTLN and CMLLR.

The next important result is that it is possible to reduce the computation time in the first recognition pass. For example, when we employed the reduced 50k lexicon, the recognition took only 59 % of real time and although the WER was 21 %, the corresponding phonetic transcription was still accurate enough for estimating the CMLLR transform. The final WER after the second recognition pass (14.8 %) was near the same as the one achieved with the 315k lexicon (14.7 %).

The reason is that Czech is an inflective language with many words forms, which often differ in only one or several characters in the prefix, suffix or word ending. Therefore during recognition with the limited lexicon, the missing word is often replaced or even entirely composed from several similar or short words: the increase in WER by 4.4 % for the 50k lexicon corresponds to increase in phoneme error rate only by 1.8 %.

The last performed experiment (Tab. 5) shows the detail results of the complete two-step approach (with the 50k lexicon in the first pass) for various speech archives. We can see that WER was reduced in all cases and that its biggest relative reduction was achieved for the parliament recordings. The reasons are twofold. Firstly, the average length of the utterances was 50 seconds so that CMLLR transforms could be estimated properly. Secondly, the baseline WER was the lowest one, because we employed the general newspaper LM, which was not tuned for the parliament domain.

Table 5: *Final results for different spoken archives*

	radio news	**talk-shows**	**TV news**	**parliament speeches**
# **of words**	16590	17461	38255	22014
SI WER [%]	17.3	18.2	14.6	23.4
SA WER [%]	13.8	15.6	12.7	18.5
rel. reduction of WER [%]	20.2	14.3	13.0	21.0

6. Conclusion

In this paper, several unsupervised feature-based SA techniques were investigated and evaluated experimentally for transcription of various speech archives including radio and TV news, talk-shows and parliament speeches. The resulting two-step adaptation approach yielded significant decrease in WER for all these data. It should also be noted that all the experiments were performed for manually created speech segments and that an automatic segmentation algorithm is employed in practice.

7. Acknowledgements

This work was supported by the Czech Science Foundation - GACR (grant no. P103/11/P499), by the Technology Agency of the Czech Republic (project no. TA01011142) and by Czech Ministry of culture (project NAKI no. DF11P01OVV013).

8. References

[1] J. H. L. Hansen et al., "SpeechFind: advances in spoken document retrieval for a National Gallery of the Spoken Word," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 712–730, Sep. 2005.

[2] W. Byrne et al., "Automatic recognition of spontaneous speech for access to multilingual oral history archives," *Speech and Audio Processing, IEEE Transactions on*, vol. 12, no. 4, pp. 420–435, 2004.

[3] J. Nouza, J. Zdansky, P. Cerva, and J. Kolorenc, "A System for Information Retrieval from Large Records of Czech Spoken Data," in *Text, Speech and Dialogue*, 2006, vol. 4188, pp. 485–492.

[4] C. Huang, T. Chen, and E. Chang, "Adaptive model combination for dynamic speaker selection training," in *INTERSPEECH*, 2002.

[5] T. Anastasakos, J. Mcdonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *Proc. ICSLP*, 1996, pp. 1137–1140.

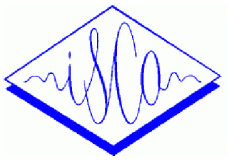
[6] L. Welling, H. Ney, S. Kanthak, and L. F. I. Vi, "Speaker adaptive modeling by vocal tract normalization," *IEEE Trans. on Speech and Audio Processing*, vol. 10, pp. 415–426, 2002.

[7] S. Molau, S. Kanthak, and H. Ney, "Efficient vocal tract normalization in automatic speech recognition," in *Proc. of the ESSV00*, 2000, pp. 209–216.

[8] M. J. F. Gales, "Maximum Likelihood Linear Transformations for HMM-Based Speech Recognition," *Computer Speech and Language*, vol. 12, pp. 75–98, 1998.

[9] A. Vandecatseye et al., "The COST278 pan-European broadcast news database," in *Proceedings LREC 2004*, 2004, pp. 873–876.

ISCA Archive
http://www.isca-speech.org/archive



INTERSPEECH 2012
ISCA's 13th Annual Conference
Portland, OR, USA
September 9-13, 2012

Study on Integration of Speaker Diarization with Speaker Adaptive Speech Recognition for Broadcast Transcription

Jan Silovsky, Petr Cerva, Jindrich Zdansky and Jan Nouza

Institute of Information Technology and Electronics, Faculty of Mechatronics,
Technical University of Liberec, Liberec, Czech Republic
{jan.silovsky,petr.cerva,jindrich.zdansky,jan.nouza}@tul.cz

Abstract

In this paper we study a close incorporation of speaker diarization with speaker adaptive speech recognition in our broadcast transcription system. We provide our motivation for utilization of speech transcripts in the diarization process and analyze the effect it yields in terms of diarization performance or computational cost. Further, speaker adaptation performed according to various scenarios of speaker segmentation and diarization of an audio stream is evaluated. For better insight, the limit performance is evaluated substituting most of the components of the system by the oracle ones.

Index Terms: Speaker diarization, i-vectors, speaker adaptation, CMLLR, broadcast transcription

1. Introduction

Unsupervised speaker adaptation is considered a standard part of most of complex speech recognition systems, e.g. those developed by various sites participating in the EARS, GALE, AMI, CHIL programs and others. Usually speaker adaptation techniques require a knowledge of a preliminary transcript of the adaptation data and hence multiple recognition passes are performed. Further, a knowledge about attribution of temporal segments of a continuous stream to appropriate originating speakers is required. Speaker diarization is hence a useful preprocessing step with respect to speaker adaptation as it aims to determine the number of speakers as well as their occurrence in the given audio stream [1, 2].

On the other side, speech recognition could be a useful preprocessing step for speaker diarization [3]. In this paper we aim to analyze the effect that a knowledge of speech transcripts yields to the speaker diarization process and the effect speaker diarization in return yields to speaker adaptive (SA) speech recognition. To gain better insight, various components of the system are substituted by *oracle* components. By which we refer to utilization of human-produced reference annotations (both speaker segmentation and speech transcripts) instead of automatic outputs provided by the components.

In our system, we make use of information about word boundaries and we also take advantage of classification of various non-speech events as provided by our automatic speech recognition (ASR) system. This allows the diarization system to neglect various noises produced by speakers (breathing, various hesitation sounds, cough, lip-smack, etc.) that carry no speaker-specific information (considering representation of the signal by cepstral features) and thus harm the representation of clusters. Moreover, distribution of the cepstral features corresponding to these sounds differs notably from the distribution corresponding to speech regions and hence a false change-point is often detected due to high values of segmentation test statistics. However, at the same time, these sounds should not break the continuity of segments uttered by a single speaker and should thus not be simply dismissed. Although linguistic information contained in transcripts could provide further cues for discrimination between speakers [4], we avoid its use as it would make the speaker diarization system language dependent. To show how the transcripts are treated, in the next section we present a detail description of our speaker diarization system.

2. Speaker diarization system

The traditional framework of a speaker diarization system consists of three basic modules [5] that perform tasks of speech activity detection (SAD), speaker change-point detection and speaker clustering. In our system, availability of speech transcripts allows us to completely dismiss the output of the standard SAD module (e.g. energy or model based).

2.1. Utilization of speech transcripts

The non-speech events are categorized into two classes depending on whether they were produced by speakers (breathing, various hesitation sounds, cough, lip-smack, etc.) or they are artificial (e.g. music, background noise). Let us denote these classes $\mathcal{C}_h$ and $\mathcal{C}_n$ respectively. In addition, all lexicon entries (words) are comprised in the class $\mathcal{C}_s$. Further, let $\boldsymbol{p} = \{p_1, \ldots, p_M\}$ be a sequence of times corresponding to start position of transcription elements w_i $(i = 1, \ldots, M)$ in a transcript produced by the ASR module. For a stream represented by a sequence of T feature vectors, we then introduce two binary sequences $a_h(t)$ and $a_n(t)$ where $t = 1, \ldots, T$. Values of $a_h(t)$ and $a_n(t)$ are given as

$$a_h(t) = \begin{cases} 1 & \text{if } w_i \in \{\mathcal{C}_s, \mathcal{C}_h\} \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad a_n(t) = \begin{cases} 1 & \text{if } w_i \in \mathcal{C}_s \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $p_i \leq t < p_{i+1}$ and $i = 1, \ldots, M$.

Let us highlight that we do not use the information about speech activity (as determined by the classification of transcript elements) to split the signal but all frames (accompanied by the $a_h(t)$ and $a_n(t)$ values) of the stream are passed to the speaker segmentation step. Hence, the speaker homogeneous segments interleaved by non-speech intervals are not broken regardless of the kind and duration of these intervals. Smoothing of short speech or non-speech intervals that break the fluency of the output is postponed at the end of the diarization process.

The fact that the diarization process does not rely on lexical content of transcripts brings the advantage of lower dependency on language of the ASR module.

2.2. Speaker segmentation

Let $\mathcal{O} = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_T\}$ be a sequence of d-dimensional feature vectors (frames) representing an audio stream. The sequence is traversed by a sliding variable-length window and a frame within the window is claimed to be a change-point based upon a test of hypotheses. Hypotheses testing requires derivation of proper test statistics. We use the test statistic derived based on the maximum likelihood approach [6] defined as

$$\Lambda(t) = \alpha\sqrt{n \log|\boldsymbol{\Sigma}| - n_1 \log|\boldsymbol{\Sigma}_1| - n_2 \log|\boldsymbol{\Sigma}_2|} - \beta \quad (2)$$

where n_1 and n_2 denote the number of effective frames in the left and right partition of the analyzed window as split by a hypothesized change-point at the time t, $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ are full covariance matrices corresponding to these partitions and $\boldsymbol{\Sigma}$ is a full covariance matrix of all effective frames in the analyzed window. Finally, the remaining terms in (2) are defined as[1]

$$\alpha = (2 \log\log n)^{1/2}, \quad (3)$$

$$\beta = 2 \log\log n + d \log\log\log n - \log\Gamma(d). \quad (4)$$

The information about the start position of transcript elements is used as the constraining condition for choice of change-point candidates. Hence a single change-point candidate within the analysis window is found according to the equation

$$\hat{t} = \underset{d<t<n-d}{\operatorname{argmax}} R(t), t \in \boldsymbol{p}. \quad (5)$$

Let us remark that as we are using automatic transcripts, this does not mean that a change-point cannot be detected within a word actually uttered by a speaker, but the chance of such detection is significantly decreased. When $\Lambda(\hat{t})$ exceeds a given decision threshold θ, the change-point at the time $\hat{t}$ is confirmed. The decision threshold is set so as to prefer over-segmentation to miss of change-points (the threshold is lower than the optimal threshold found by the training algorithm described in [6]). This is preferable as false detections may be eliminated in the clustering stage while missed change-points are unrecoverable.

The covariance matrix (see eq. (2)) for a partition of the stream ranging from the time t_1 to t_2 is computed based on the first-and second-order statistics as follows

$$\boldsymbol{\Sigma} = \frac{1}{n}\Delta\boldsymbol{S} - \frac{1}{n^2}\Delta\boldsymbol{F}(\Delta\boldsymbol{F})^T \quad (6)$$

where $n = \sum_{t=t_1}^{t_2} a_n(t)$, $\Delta\boldsymbol{F} = \sum_{t=t_1}^{t_2} a_n(t)\boldsymbol{o}_t$ and $\Delta\boldsymbol{S} = \sum_{t=t_1}^{t_2} a_n(t)\boldsymbol{o}_t\boldsymbol{o}_t^T$. In practice, efficient computation of the $\Delta\boldsymbol{F}$ and $\Delta\boldsymbol{S}$ is achieved by differentiation of first-and second-order statistics iteratively accumulated in a circular buffer [6].

2.3. Speaker clustering

Our clustering module uses bottom-up clustering. More specifically, we employ the two-stage clustering scenario described in [7]. At the first pre-clustering stage, the similarity of clusters is measured via the standard criterion based on the Bayesian Information Criterion (BIC) difference. At the second stage, clusters are represented by *i-vectors* and their similarity is measured by their cosine distance [8].

In the i-vector concept, a simple factor analysis model is employed to extract a fixed-and low-dimensional representation of a segment of variable length in the *total variability space* (TVS) [8]. A projection from a sequence of feature vectors $\mathcal{O} = \{\boldsymbol{o}_1, \ldots \boldsymbol{o}_{T_s}\}$ representing an audio segment to TVS is provided by computation of a Maximum A Posterior (MAP) point estimate of the so called *i-vector* $\boldsymbol{x}$ based on the zero-and first-order sufficient statistics gathered employing a Gaussian Mixture Model (GMM) as follows [9]:

$$\boldsymbol{x} = \left(\boldsymbol{I} + \sum_{c=1}^{C} N_c \boldsymbol{T}_c^T \boldsymbol{\Sigma}_c^{-1} \boldsymbol{T}_c\right)^{-1} \sum_{c=1}^{C} \boldsymbol{T}_c^T \boldsymbol{\Sigma}_c^{-1} \tilde{\boldsymbol{F}}_c(\boldsymbol{m}_c) \quad (7)$$

where $\boldsymbol{T}$ is a low-rank rectangular matrix representing the total variability space which can be decomposed into $\boldsymbol{T}_c$ blocks so that $\boldsymbol{T} = \left[\boldsymbol{T}_1^T \ldots \boldsymbol{T}_C^T\right]^T$, $\boldsymbol{m}_c$ and $\boldsymbol{\Sigma}_c$ are a mean vector and a diagonal covariance matrix corresponding to the c-th component of the GMM (having C components in total) respectively. Finally, the zero-and (centralized) first-order statistics are computed respectively as follows

$$N_c = \sum_{t}^{T_s} \gamma_c(t) a_n(t) \quad (8)$$

$$\tilde{\boldsymbol{F}}_c = \sum_{t}^{T_s} \gamma_c(t) a_n(t)(\boldsymbol{o}_t - \boldsymbol{m}_c) \quad (9)$$

where $\gamma_c(t)$ is the posterior probability of the event that feature vector $\boldsymbol{o}_t$ is accounted for by the c-th component of the GMM. This way of treatment of transcripts thus results in a form of a frame level purification [10].

Having the i-vector representation of segments (or clusters) g_1 and g_2 by i-vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ respectively, we can assess their similarity simply using the cosine distance as

$$d(g_1, g_2) = \frac{\boldsymbol{x}_1^T \boldsymbol{x}_2}{\|\boldsymbol{x}_1\| \, \|\boldsymbol{x}_2\|}. \quad (10)$$

The higher the cosine distance is, the more likely both segments originate from the same speaker. In the clustering process, the two clusters with the highest cosine distance score are merged together and if a maximum value of the score for any pair of clusters drops under a given threshold, the stopping condition of the process is met.

Because Eq. (10) can be applied only for a pair of i-vectors, the i-vector representing a certain cluster is simply computed as an *average* of all i-vectors corresponding to the initial segments assigned to the given cluster so far [11]. The Linear Discriminant Analysis (LDA) is employed to cope with the nuisance intra-speaker variability.

2.4. Diarization output smoothing

The aim of the smoothing is to discard either short speech or non-speech intervals that harm the fluency of the output as high granularity of the output is very obtrusive to users.

While the segmentation and clustering modules use $a_n(t)$ values to distinguish between the speech and non-speech frames, here $a_h(t)$ values are taken into account. Speech and non-speech segments shorter than 0.25 s and 1.0 s respectively are discarded (in the order as listed).

3. Speech recognition system

We use our Czech LVCSR system to obtain speech transcripts. The core of the system is formed by one-pass speech decoder performing time-synchronous Viterbi search. The acoustic model was discriminatively trained using the Minimum Phone

[1] where $\Gamma(.)$ is the gamma function, i.e. $\Gamma(n) = (n-1)!$.

Error (MPE) criteria on a database containing 300 hours (from 3940 male and 2030 female speakers) of broadcast and close-talk microphone recordings. The model consists of tied-state context dependent HMMs of Czech phonemes and several types of non-speech events. It contains 4k of physical states with up to 32 components per state so that the total number of Gaussian components is approximately 120k.

The lexicon of the system contains 315k items, many of which have multiple pronunciations. The language model is based on bigrams estimated on a 10 GB corpus compiled from Czech (mainly newspaper) texts. We apply the modified Kneser-Ney discounting method for smoothing of the model.

3.1. Speaker adaptation

We use the Constrained Maximum Likelihood Linear Regression (CMLLR) for speaker adaptation. The method computes a feature transformation that aims to reduce the mismatch between an initial acoustic model and the adaptation data. Estimation of adaptation parameters consists of an iterative process where the likelihood of the adaptation data is being maximized with respect to a given preliminary transcription (which is the result of the first recognition pass). We apply only global CMLLR transformation for all Gaussians of the acoustic model. The non-speech events, as determined by the first recognition pass (taking into account values a_n), are left out from the adaptation data. During the second recognition pass, adapted speaker-specific models are applied according to the smoothed diarization output. Segments attributed to any speaker are recognized with the initial speaker independent model.

4. Experiments and results

4.1. Datasets

Experiments were carried out using two test sets. The first set was created based on the COST278 multilingual pan-European broadcast news database. This set was used in order to demonstrate the effect of utilization of speech transcripts within the speaker diarization process even in the case of language mismatch of the ASR system. The second set was composed of recordings of Czech broadcast programs only. Training and development of the diarization system was common for both test sets and it was done using the held out part of the COST278 database. Utilization of the training data and development data followed the setup described in [7].

The COST278 test data comprised of 5 languages: Belgian Dutch, Czech, Hungarian, Slovak and Slovenian. The set consisted of 15 TV news shows of length in the range from 4 to 53 minutes (6.3 hours in total). The number of speakers in these shows varies in wide range from 2 up to 78.

The Czech test set consisted of 16 shows of length in the range from 4 to 51 minutes (4.3 hours in total). These recordings encompass both TV and radio programs. The number of speakers in these shows varies in range from 2 up to 55. In contrast to the first test set, the second set contains not just news but also interviews and thus more recordings contain less than 5 speakers in this case.

4.2. Evaluation metrics

Performance of diarization systems is usually evaluated by the Diarization Error Rate (DER). The DER can be decomposed as DER = SPKE + FA + MISS, where the SPKE, FA and MISS represent the *speaker*, *speech false alarm* and *missed speech* error rates respectively. A forgiveness region of 0.25 s (both + and -) was not scored around each boundary.

We also evaluate our system in terms of standard measures used to assess the segmentation performance. For their assessment, change-points detected by the system must be coupled with the reference ones first. A *couple* is constituted iff the detected change-point is the closest to the reference one and vice versa and, in addition, if the distance between them is smaller than 1 second. Then the *recall*, the *precision* and the *F-rate* measures are calculated respectively as

$$R = \frac{H}{H+D}, \quad P = \frac{H}{H+I} \quad \text{and} \quad F = \frac{2RP}{R+P} \qquad (11)$$

where H, I, D denote the number of *coupled*, *inserted* and *deleted* change-points respectively.

Additionally, in order to highlight the effect of utilization of speech transcripts to improve the accuracy of positions of detected speaker turns, we assess the ratio of change-points that were detected within the intervals corresponding to the words uttered by speakers. We denote this ratio as the *word-breakage* (WB) rate. A forgiveness region of only 20 ms around the word boundaries was used in this case.

Finally, speech recognition performance is evaluated in terms of the *word-error-rate* (WER).

4.3. Diarization system setup

The diarization system operates with feature vectors formed by 12 Mel-Frequency Cepstral Coefficients (MFCCs). The universal background GMM with 256 Gaussians was employed for extraction of sufficient statistics. We used 400-dimensional i-vectors and the LDA dimensional reduction to 200.

The UBM was trained using the data from 1007 speakers (2530 segments, 11.5 hours). The total variability space was estimated using a subset of the UBM training data resulting from the condition of minimal length of a segment of 3 seconds. This resulted in 2050 segments (10.2 hours) from 909 speakers. The LDA projection matrix was estimated using the data from speakers for which at least three segments of minimal length of 3 seconds are available, in total 1528 segments (7.5 hours) from 280 speakers were used.

4.4. Results

Tabs. 1 and 2 present achieved results in terms of both speaker segmentation and diarization performance. The system that makes no use of transcripts employs a SAD module that combines an energy and model (GMM) based detection. We point out that the segmentation measures were evaluated at the end of the diarization process (after the clustering stage). We conclude that utilization of speech transcripts in the diarization process yields improvement of both segmentation and clustering performance measures as determined by the higher F-rate and lower DER respectively. We can also conclude that incorporation of information about word boundaries yields remarkable reduction of the WB rate. Moreover, reduction of the number of change-point candidates leads to lower computational cost[2] of the segmentation process as recognized by the real-time (RT) factor[3] in the Tab. 1. In our experiments, the knowledge of oracle transcripts yields further slight improvement of the segmentation

[2]Please note that as the application of speaker adaptive recognition requires two speech recognition passes, the first recognition pass preceding the diarization process does not yield any extra burden.

[3]Measured on a machine with Intel Core i7@2.66GHz.

Table 1: *Segmentation and diarization performance for the multilingual COST278 test set. The WB rate was evaluated only for the Czech part of the set.*

	Change-point detection					Diarization				
Transcripts	R [%]	P [%]	F [%]	WB [%]	× RT	MISS [%]	FA [%]	SPKE [%]	DER [%]	× RT
no	87.5	53.8	66.6	49.9	0.14	1.8	0.6	11.5	13.9	0.05
yes	80.1	74.6	**77.2**	**6.5**	**0.02**	2.4	0.7	**8.4**	**11.5**	0.04

Table 2: *Segmentation and diarization performance for the Czech test set.*

	Change-point detection				Diarization			
Transcripts	R [%]	P [%]	F [%]	WB [%]	MISS [%]	FA [%]	SPKE [%]	DER [%]
no	83.2	52.8	64.6	42.3	2.9	3.2	15.6	21.7
yes	78.4	73.1	**75.7**	**10.5**	3.3	3.0	13.2	**19.5**
oracle	80.2	76.0	78.0	0.0	4.0	2.7	13.3	20.0
oracle	oracle				4.0	0.0	10.2	14.2

Table 3: *Results of the SA system for the Czech part of the COST278 test set. The SI system achieved WER of 14.46 %*

Segmentation	Clustering	WER [%]	rel. impr. [%]
fixed-blocks	-	14.27	1.3
baseline	-	13.82	4.4
ASR-based	-	**13.73**	5.0
oracle	-	13.62	5.8
baseline	baseline	**13.26**	8.3
ASR-based	ASR-based	13.49	6.7
oracle	oracle	13.12	9.3

Table 4: *Results of the SA system for the Czech test set. The SI system achieved WER of 24.41 %*

Segmentation	Clustering	WER [%]	rel. impr. [%]
fixed-blocks	-	23.32	4.5
baseline	-	23.12	5.3
ASR-based	-	**22.69**	7.0
oracle	-	22.44	8.1
baseline	baseline	22.07	9.6
ASR-based	ASR-based	22.08	9.5
oracle	oracle	22.01	9.8

performance but the overall diarization performance is not affected by this knowledge. Finally, Tab. 2 provides results for the case of true speaker change-points available to the system.

Tabs. 3 and 4 summarize the speech recognition performance of the SA systems. First, adaptation for speaker homogeneous segments found by the speaker change-point detection was evaluated and compared with the adaptation performed for fixed-length blocks (results for blocks containing 5 minutes of speech are reported here). Next, adaptation for clusters defined by the diarization output was assessed. As expected, the latter scenario yields better results. In both scenarios, the performance achieved using automatic segmentation or diarization techniques was close to the performance achieved by systems using the oracle components.

5. Conclusions

In this paper we have studied the effect of utilization of automatic transcripts within the speaker diarization task as well as utilization of the speaker diarization output with respect to the speaker adaptive speech recognition. We conclude that utilization of the transcripts in the diarization process yields improvement in terms of both speaker segmentation and overall diarization performance. Further, it reduces computational cost of the segmentation process. As expected, we have demonstrated that utilization of the diarization output for the speaker adaptation outperforms adaptation performed for fixed-length blocks as well as adaptation for speaker homogeneous segments as identified by the speaker change-point detection. If the speech recognition accuracy is of the only interest, it is not important whether the automatic transcripts are taken into account during the diarization process or not. However, the main reason for utilization of transcripts is improvement of the quality of the diarization output.

6. Acknowledgments

This research work was supported by Czech Ministry of Culture (project no. DF11P01OVV013 in program NAKI).

7. References

[1] J.-L. Gauvain, L. Lamel, and G. Adda, "Partitioning and transcription of broadcast news data," in *ICSLP'98*, 1998, pp. 1335–1338.

[2] A. Stolcke, G. Friedland, and D. Imseng, "Leveraging speaker diarization for meeting recognition from distant microphones," in *ICASSP 2010*, march 2010, pp. 4390 –4393.

[3] J. Huang, E. Marcheret, K. Visweswariah, and G. Potamianos, "The IBM RT07 evaluation systems for speaker diarization on lecture meetings," in *Multimodal Technologies for Perception of Humans*, ser. LNCS, 2008, vol. 4625, pp. 497–508.

[4] L. Canseco-Rodriguez, L. Lamel, and J.-L. Gauvain, "Speaker diarization from speech transcripts," in *INTERSPEECH*, Jeju Island, Korea, October 2004.

[5] S. Tranter and D. Reynolds, "An overview of automatic speaker diarization systems," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 5, pp. 1557–1565, 2006.

[6] J. Zdansky, "BINSEG: An efficient speaker-based segmentation technique," in *INTERSPEECH*, Pittsburgh, PA, USA, September 2006.

[7] J. Silovsky and J. Prazak, "Speaker diarization of broadcast streams using two-stage clustering based on i-vectors and cosine distance scoring," in *ICASSP*, Kyoto, Japan, March 2012, pp. 4193–4196.

[8] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 4, pp. 788 –798, May 2011.

[9] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *IEEE Transactions on Speech and Audio Processing*, vol. 13, pp. 345–359, May 2005.

[10] X. Anguera, C. Wooters, and J. Hernando, "Purity algorithms for speaker diarization of meetings data," in *ICASSP 2006*, vol. 1, may 2006, p. I.

[11] J. Franco-Pedroso, I. Lopez-Moreno, D. Toledano, and J. Gonzalez-Rodriguez, "ATVS-UAM system description for the audio segmentation and speaker diarization Albayzin 2010 evaluation," in *FALA 2010*, November 2010, pp. 415–417.

Available online at www.sciencedirect.com

ScienceDirect

Speech Communication 55 (2013) 1033–1046

SPEECH COMMUNICATION

www.elsevier.com/locate/specom

Speaker-adaptive speech recognition using speaker diarization for improved transcription of large spoken archives

Petr Cerva *, Jan Silovsky, Jindrich Zdansky, Jan Nouza, Ladislav Seps

Institute of Information Technology and Electronics, Technical University of Liberec, Studentska 2, 461 17 Liberec, Czech Republic

Received 25 March 2013; received in revised form 27 June 2013; accepted 29 June 2013
Available online 8 July 2013

Abstract

This paper deals with speaker-adaptive speech recognition for large spoken archives. The goal is to improve the recognition accuracy of an automatic speech recognition (ASR) system that is being deployed for transcription of a large archive of Czech radio. This archive represents a significant part of Czech cultural heritage, as it contains recordings covering 90 years of broadcasting. A large portion of these documents (100,000 h) is to be transcribed and made public for browsing. To improve the transcription results, an efficient speaker-adaptive scheme is proposed. The scheme is based on integration of speaker diarization and adaptation methods and is designed to achieve a low Real-Time Factor (RTF) of the entire adaptation process, because the archive's size is enormous. It thus employs just two decoding passes, where the first one is carried out using the lexicon with a reduced number of items. Moreover, the transcripts from the first pass serve not only for adaptation, but also as the input to the speaker diarization module, which employs two-stage clustering. The output of diarization is then utilized for a cluster-based unsupervised Speaker Adaptation (SA) approach that also utilizes information based on the gender of each individual speaker. Presented experimental results on various types of programs show that our adaptation scheme yields a significant Word Error Rate (WER) reduction from 22.24% to 18.85% over the Speaker Independent (SI) system while operating at a reasonable RTF.

Keywords: Speaker adaptive; Automatic speech recognition; Speaker adaptation; Speaker diarization; Automatic transcription; Large spoken archives

1. Introduction

Automatic processing of large-scale spoken archives has attracted a lot of attention during the last decade. There has been a growing interest in the development of efficient methods enabling transcription and search of audio or audio-video data. Nowadays, this interesting and wide application area includes systems for transcription, indexing and retrieval of various sources.

For example, several solutions (such as Glass et al., 2007) have been developed to improve the quality and accessibility of higher education by allowing students to browse the content of (academic) lectures over Internet. Another application is the Global Autonomous Language Exploitation (GALE) program. Its main goal is to distill information from publicly available broadcast sources in multiple languages, and to make it accessible to English speakers. In the USA, a system called SpeechFind (Hansen et al., 2005) was developed to enable automatic processing, indexing and browsing of the National Gallery of the Spoken Word. Similarly in Europe, an initiative named CHoral aims to provide public access to Dutch oral history collections (Ordelman et al., 2006). The goal of the MALACH project is to enable access to multilingual oral history archives preserving the stories of survivors and witnesses of the Holocaust (Byrne et al., 2004).

We have been working on the automatic transcription task since 2005. In 2006, we presented the first Large Vocabulary Continuous Speech Recognition (LVCSR)

* Corresponding author. Tel.: +420 485353778.
E-mail addresses: petr.cerva@tul.cz (P. Cerva), jan.silovsky@tul.cz (J. Silovsky), jindrich.zdansky@tul.cz (J. Zdansky), jan.nouza@tul.cz (J. Nouza), ladislav.seps@tul.cz (L. Seps).

0167-6393/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.specom.2013.06.017

system for automatic online monitoring of Czech broadcast news (Nouza et al., 2006). Later, a more advanced system was applied for the full-text search of a collection of Czech broadcast programs (Nouza et al., 2010). One of our current projects aims at processing, indexing and accessing data collected in a large Czech Radio archive which embodies an important part of Czech cultural heritage (Nouza et al., 2012). This archive contains more than 200,000 individual recordings covering broadcasting in the Czech Republic and former Czechoslovakia since 1923. A large portion of this archive (100,000 h) should be transcribed with the best possible accuracy and then indexed to enable search.

For automatic transcription, we employ our own recognition engine designed to cope with highly inflective Czech language. Its lexicon has to contain more than 550,000 (550k) entries to assure an out-of-vocabulary (OOV) rate lower than 2% for most of the recordings. In this work, we propose an offline speaker-adaptive scheme to improve the transcription accuracy of our ASR system for the large spoken archive we mentioned above. In contrast to most existing concepts, our scheme is designed with high regard for the resulting RTF of the entire adaptation process. We try to achieve a low RTF value because the amount of data for processing is very high in our case and, as mentioned, the system has to operate with a large lexicon.

This paper is organized as follows: the next section provides an overview of related work. The entire scheme of our speech processing framework is then presented in Section 3. Section 4 details the speech decoding engine employed within this work from language and acoustic modeling point of view. An integral part of our framework is a speaker diarization module which is described in Section 5 along with the way of its integration within the framework. Section 6 deals with the development of an unsupervised speaker adaptation approach which utilizes the output from the speaker diarization module. The last Section 7 then concludes this paper.

2. Related work

The state-of-the-art systems for processing of spoken archives employ a core ASR system that converts an input spoken document to its text form. This task is obviously challenging as the used engine has to deal with a lot of various difficulties (e.g., spontaneous speech, high speaker variability, limited resources of data for language modeling, etc.).

In most of these systems, such as those developed by various sites participating in the EARS, GALE, AMI and other programs, the unsupervised SA module is considered a standard component. The reason is that it should allow for improving recognition accuracy with respect to individual speakers occurring in the given document. Usually, speaker adaptation techniques (Shinoda, 2005) require knowledge of attribution of temporal segments in a continuous stream to appropriate originating speakers. Within this process, speaker diarization (Moattar and Homayounpour, 2012) is hence a useful preprocessing step for speaker adaptation as it aims to determine the number of speakers as well as their occurrence in time within the given audio stream (Gauvain et al., 1998). Two basic principles exist for performing speaker diarization with respect to speaker adaptation.

The first concept utilizes online speaker clustering. It is suitable mainly for real-time transcription systems as it allows speech segments with corresponding speaker labels to be used almost immediately for speech recognition. This leads to only a small time delay between the speech and recognizer's output. The output from online speaker diarization can then be used for incremental speaker adaptation as in Liu et al. (2005). Breslin et al. (2011) studied a close integration of speaker adaptation and clustering based on utilization of adaptation parameters for representation of speakers within a clustering process. More specifically, the same statistics accumulated for given speech segments were used for both speaker clustering and adaptation.

The second way of integrating of speaker diarization and adaptation techniques relies on offline processing and the assumption that all data is available to the system. Speaker diarization is usually performed within this concept before adaptation and it is also a preliminary step for speech recognition (Chu et al., 2009; Hain et al., 2012). The traditional framework of an offline speaker diarization system consists of three basic modules that perform tasks of speech activity detection (SAD), speaker change-point detection and speaker clustering. The resulting diarization output then serves as the input for cluster-based batch speaker adaptation.

It should also to be noted that most speaker adaptation techniques utilize knowledge of a preliminary transcript of the data. Hence when SA methods are employed, the recognition process is usually carried out at least in two decoding passes, where the first one serves to create the transcripts automatically. Multi-pass transcription can also be employed for Language Model (LM) rescoring to further improve recognition results. For example in Chu et al. (2008), a word hypothesis lattice generated in the first decoding pass is rescored using a larger LM or an LM adapted to the output from the first pass.

Most systems for offline transcription also combine a large number of different recognition systems to achieve WER that is better than results yielded by any of the individual systems (Hain et al., 2012; Matsoukas et al., 2006; Chu et al., 2009). These should produce complementary output created, e.g., by employing different LMs, speaker adaptation scenarios, features or acoustic model training schemes.

Generally, multiple recognition outputs can be combined sequentially or in parallel. The former option is called cross-adaptation. Here, the output of one pass is employed to adapt models of another system. The latter option can be carried out using several different strategies. The ones whose use is most widespread rely on the

Recognizer Output Voting Error Reduction (ROVER) method (Fiscus, 1997) or Confusion Network Combination (CNC) (Mangu et al., 2000). For example, the AMIDA system (Hain et al., 2012) for transcription of meetings takes particular advantage of cross-adaptation, while the latter (parallel-combination) approach is used in the SRI-ICSI system (Stolcke et al., 2007) for meeting and lecture recognition. However, the disadvantage of ROVER, CNC and related approaches is that they can violate the word order of the original inputs. To overcome this limitation, a method based on direct combination of multiple lattices is proposed in Chu et al. (2009). This approach produces a time-coherent final hypothesis, which may, for example, be more appropriate for post-ASR sentence segmentation.

All the previously mentioned approaches for system combination can be employed within various architectures. It is possible, for example, to obtain ROVER output from multiple independent systems, which is then used to adapt the same or different set of systems (Matsoukas et al., 2006). The resulting RTF of the whole offline transcription process can then easily reach values of 10 or even more (Matsoukas et al., 2006; Stolcke et al., 2007).

3. Proposed speaker-adaptive scheme

The scheme of our speaker-adaptive approach extends the scheme that we presented in Silovsky et al. (2012) and it is depicted in Fig. 1. In contrast to most solutions described in the previous section, our framework is designed with a high regard for the resulting real-time factor. Therefore, we employ only two decoding passes and the output from the first SI pass is used not only for adaptation as usual but also for speaker diarization. Moreover in Section 6.7, we show that the biggest reduction of the adaptation computation time can be reached by reducing the number of items in the lexicon during the first decoding pass. The presented results prove that this reduction can be performed without the loss of final recognition accuracy in the second decoding pass and without a negative effect on results of speaker diarization.

Availability of transcripts for diarization allows us to completely dismiss the output of the standard SAD module (e.g., energy- or model-based) and speed up the diarization process. This means that we make use of information about word boundaries and we also take advantage of classification of various non-speech events as done by our ASR system. This also allows us to neglect various noises produced by speakers (breathing, various hesitation sounds, cough, lip-smack, etc.) that carry no speaker-specific information and thus harm the representation of clusters. To show how the transcripts are treated, we present a detailed description of our speaker diarization system in Section 5.

As aforementioned, the transcripts used for diarization then also serve for cluster-based speaker adaptation. Along the speaker diarization output, our speaker diarization engine provides gender classification of the identified speaker as a by-product. Hence, our adaptation approach employs information about the detected gender of each speaker to facilitate choice of an initial Gender Dependent (GD) model for adaptation. Benefits of this strategy are proven on a large development set in Section 6. The study presented within this section is focused on evaluation of various scenarios relying on different levels of speaker diarization output in combination with different adaptation scenarios.

4. Employed recognition engine

We use our own ASR system to obtain speech transcripts. The core of the engine is formed by a one-pass speech decoder performing time-synchronous Viterbi search.

The system uses an SI acoustic model based on tied-state context dependent HMMs of Czech phonemes and several types of non-speech events (e.g., breathing, various hesitation sounds, cough, lip-smack, etc.). This model contains 4k physical states with up to 32 Gaussian components per state (i.e., 120k components in total). It was trained using 300 h of speech recordings. The features are 13-dimensional MFCCs with Δ and $\Delta\Delta$. The sampling frequency used is 16 kHz.

The linguistic part of the system consists of a lexicon and a language model. In the basic configuration, the full lexicon contains 550k entries (word forms and multi-word collocations) that were observed most frequently in a 10

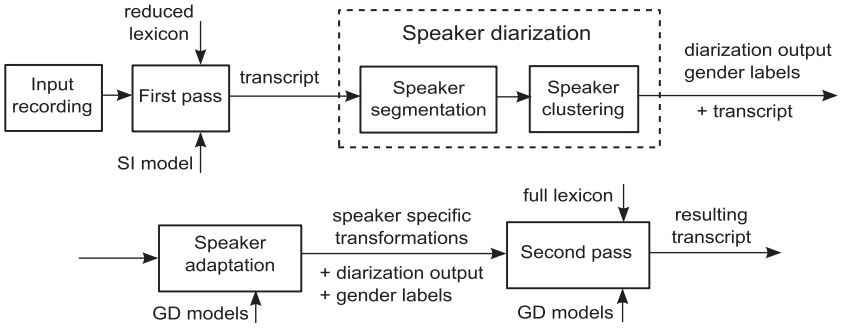


Fig. 1. The scheme of the proposed speaker adaptive ASR approach for improved transcription of large spoken archives.

GB large corpus covering newspaper texts and broadcast program transcripts. Some of the lexical entries have multiple pronunciation variants. Their total number is 580k.

The employed LM is based on N-grams. For practical reasons (mainly with respect to the very large vocabulary size), the system uses bigrams. In the training word corpus, 159 million unique word-pairs (1062 million in total) belonging to the items in the 550k lexicon were observed. However, 20 percent of all "word-pairs" actually include sequences containing three or more words, as the lexicon contains 4k multi-word collocations. The unseen bigrams are backed-off by the Kneser-Ney smoothing technique (Kneser and Ney, 1995).

5. Speaker diarization module

Our speaker diarization scenario follows a standard framework consisting of three consecutive steps. These are (a) voice activity detection, (b) speaker turn detection and (c) speaker clustering.

While voice activity detection is often considered as mature, the shift of interest from controlled and clean conditions to real-life and more diverse environments and application domains (Ng et al., 2012) reveals that this is still an open field for further research efforts. In our experience, ASR transcripts provide better cues to judge voice activity compared to other techniques, e.g., those based on Gaussian Mixture Models (GMMs). Apparently, the computational cost of obtaining an ASR transcript exceeds the cost of most other SAD techniques. However, since we are dealing with multi-pass speaker-adaptive scenario, we have ASR transcripts at our disposal at no extra cost after the first recognition pass. Hence, we rely on these transcripts in our speaker diarization framework.

Once speech activity is resolved, speaker turn detection is performed. Given the representation of a stream by a sequence of feature vectors (frames), the sequence is traversed by a sliding variable-length window. A frame within the window is proclaimed to be a speaker change-point if a model selection criterion favors the model representing the two parts of the window, as separated by the investigated frame, by two individual distributions over the model representing all data with the aid of a single distribution.

Speaker clustering then aims to group together speaker-homogeneous segments presumed from the same speaker. We employ a two-stage bottom-up clustering scenario presented in Silovsky and Prazak (2012). At the first clustering stage, the similarity of clusters is measured via the standard criterion based on the Bayesian Information Criterion (BIC) difference (Chen and Gopalakrishnan, 1998). In the second stage, clusters are represented by *i-vectors* and their similarity is measured by their cosine similarity (Dehak et al., 2011). Finally, smoothing is applied to diarization output as will be described later.

5.1. Utilization of speech transcripts

Speech transcripts are utilized in two different ways. First, as mentioned above, they provide robust discrimination between speech and non-speech frames. Second, knowledge of transcripts allows us to refine speaker turn detection as well as reduce its computational cost. This is achieved by using information about start positions of transcription elements as a constraining condition for choice of change-point candidates.

In Silovsky et al. (2012) we demonstrated that utilization of ASR transcripts yielded improvement of speaker turn detection performance in terms of the F-rate relatively by 16% (from 66.6% to 77.2%). At the same time, the RTF associated with speaker segmentation was reduced from 0.14 to 0.02. Further, quality of diarization output was assessed by the ratio of speaker change-points that were detected inside regions corresponding to elements of *reference* (true) transcription and not at their boundaries (with a tolerance region of 20 ms). This rate was reduced from 44.9% to 6.5%.

Let us emphasize that we do not use the information about speech activity to split the signal into speech and non-speech segments, nor do we apply any smoothing to the speech activity detection output. Instead, all frames representing the stream are passed to the subsequent processing steps accompanied by the corresponding speech activity labels. Hence, the speaker homogeneous segments interleaved by non-speech intervals are not broken regardless of duration of these intervals. Smoothing of short speech or non-speech intervals that harm fluency of the output is postponed at the end of diarization process.

Non-speech events modeled by our acoustic model can be categorized into two classes depending on whether they were produced by speakers (breathing, various hesitation sounds, cough, lip-smack, etc.) or they are artificial (e.g., music, background noise). Within speaker segmentation and clustering, both types are treated equally as they are both supposed to carry no, or just a little, speaker-characteristic information and thus harm a representation of speakers. However, differentiation of non-speech events is beneficial for diarization output smoothing, since non-speech events produced by speakers are not supposed to break speaker homogeneous segments into multiple ones. Hence, non-speech events originated by speakers are considered as parts of speech for the purposes of output smoothing.

5.2. Speaker segmentation

The sequence is traversed by a sliding variable-length window and a frame within the window is claimed to be a change-point if the test statistic value exceeds a given threshold. In our system we use the test statistic based on the maximum likelihood approach (Zdansky, 2006) defined as

$$\Lambda(t) = a\sqrt{N\log|\mathbf{\Sigma}| - N_1\log|\mathbf{\Sigma}_1| - N_2\log|\mathbf{\Sigma}_2|} - b \quad (1)$$

where N_1 and N_2 denote the number of frames in the left and right partitions of the analyzed window as split by a hypothesized change-point at time t, $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$ are full covariance matrices corresponding to these partitions and $\mathbf{\Sigma}$ is the full covariance matrix of all $N = N_1 + N_2$ frames in the analyzed window. In all cases, these partitions are considered to be formed only from frames labeled as speech.

Finally, the remaining terms in (1) are defined as

$$a = (2\log\log N)^{1/2}, \quad (2)$$

$$b = 2\log\log N + d\log\log\log N - \log\Gamma(d). \quad (3)$$

where d is the dimension of feature vectors and $\Gamma(.)$ is the gamma function.[1]

The information about the start positions of transcript elements is used as a constraining condition for choice of change-point candidates. Having $\mathbf{p} = \{p_1, \ldots, p_M\}$ a sequence of times corresponding to these start positions, a single change-point candidate within the analyzed window spanned from time t_0 is found according to equation

$$\hat{t} = \underset{t_0<t<t_0+N}{\operatorname{argmax}} \Lambda(t), t \in \mathbf{p}. \quad (4)$$

Use of automatic transcripts does not ensure that a change-point cannot be detected within a word actually uttered by a speaker, but the chance of such detection is significantly reduced. When $\Lambda(\hat{t})$ exceeds a given decision threshold, the change-point at time $\hat{t}$ is confirmed. The decisive threshold is set so as to prefer over-segmentation as opposed to missing change-points. It is achieved by setting the threshold below the optimal threshold found using the training algorithm described in Zdansky (2006). This is preferable because false detections may be eliminated in the clustering stage, while missed change-points are unrecoverable.

If no change-point is detected within the current analysis window, the window is expanded by one transcript element. Once the length of the analyzed window reaches the maximum length (determined by a given duration), the start position of the window is shifted forward. On the contrary, if a change-point is detected, its location $\hat{t}$ is stored and the partition between the start of the analyzed window and the location $\hat{t}$ is searched for another speaker turn. If there is no turn in the partition, a new window encompassing two transcript elements is initialized, starting at time $\hat{t}$.

5.3. Speaker clustering

Scheme of our two-stage bottom-up clustering scenario is depicted in Fig. 2. At the first stage, the similarity of clusters is measured via the standard criterion based on the Bayesian Information Criterion (BIC) difference. This stage is intended to pre-cluster the segments with goal of reducing of the number of very short segments. At the second stage, clusters are represented by *i-vectors* and their similarity is measured by their cosine distance.

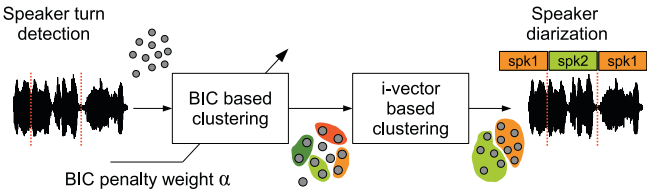


Fig. 2. The employed two-stage speaker clustering process.

The BIC-based criterion compares BIC statistics of clusters g_1 and g_2 with the BIC statistic of a cluster g formed by merging of clusters g_1 and g_2. We apply a local BIC measure which is defined as (Chen and Gopalakrishnan, 1998)

$$\Delta\mathrm{BIC}(g_1, g_2) = N\log|\mathbf{\Sigma}| - N_1\log|\mathbf{\Sigma}_1| - N_2\log|\mathbf{\Sigma}_2| - \alpha P \quad (5)$$

where N is the number of frames, $\mathbf{\Sigma}$ is the full covariance matrix (only speech frames are taken into account) of the data, α is a penalty weight, and finally, P is the penalty

$$P = \frac{1}{2}\left(d + \frac{1}{2}d(d+1)\right)\log N \quad (6)$$

where d is the dimension of feature vectors. In the clustering process, two clusters with the lowest[2] mutual ΔBIC value are merged together. If a minimal distance between any pair of clusters is higher than a certain threshold (zero in our case), the stopping criterion is met and the remaining clusters are passed to the second clustering stage. Keeping the decision threshold fixed, the switch-point between the first and second stages is controlled by the BIC penalty weight α. A smaller penalty weight invokes less compensation for bigger clusters and thus leads to an earlier switch between stages.

In the i-vector concept, a simple factor analysis model is employed to extract a fixed-and low-dimensional representation of a segment of variable length N in a *total variability space* (Dehak et al., 2011). A projection from a sequence of feature vectors $\mathcal{O} = \{\mathbf{o}_1, \ldots \mathbf{o}_N\}$ representing an audio segment to an i-vector space is provided by computation of a Maximum A Posterior (MAP) point estimate of an i-vector $\mathbf{x}$ based on the zero- and first-order sufficient statistics gathered employing a Gaussian Mixture Model (GMM) as follows (Kenny et al., 2008)

$$\mathbf{x} = \left(\mathbf{I} + \sum_{c=1}^{C} N_c\mathbf{T}_c^T\mathbf{\Sigma}_c^{-1}\mathbf{T}_c\right)^{-1}\sum_{c=1}^{C}\mathbf{T}_c^T\mathbf{\Sigma}_c^{-1}\tilde{\mathbf{F}}_c(\mathbf{m}_c) \quad (7)$$

where $\mathbf{T}$ is a low-rank rectangular matrix representing the total variability space which can be decomposed into $\mathbf{T}_c$ blocks so that $\mathbf{T} = \left[\mathbf{T}_1^T \ldots \mathbf{T}_C^T\right]^T$, $\mathbf{m}_c$ and $\mathbf{\Sigma}_c$ is a mean vector and a diagonal covariance matrix corresponding to the c-th

[1] $\Gamma(n) = (n-1)!$.

[2] ΔBIC values are usually negative, thus when referring to the lowest value we mean the value nearest minus infinity.

component of the GMM (having C components in total). Finally, the zero- and (centralized) first-order statistics are computed respectively as follows

$$N_c = \sum_{t}^{N} \gamma_c(t) \tag{8}$$

$$\tilde{\mathbf{F}}_c = \sum_{t}^{N} \gamma_c(t)(\mathbf{o}_t - \mathbf{m}_c) \tag{9}$$

where $\gamma_c(t)$ is the posterior probability of the event that feature vector $\mathbf{o}_t$ is accounted for by the GMM's component c. Again, clusters are considered to be formed only from speech frames, resulting in the form of a frame level purification (Anguera et al., 2006).

Having the i-vector representation of segments (or clusters) g_1 and g_2 by i-vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ respectively, we can assess their cosine similarity simply as

$$d(g_1, g_2) = \frac{\mathbf{x}_1^T \mathbf{x}_2}{\|\mathbf{x}_1\| \|\mathbf{x}_2\|}. \tag{10}$$

The higher the cosine similarity is, the more likely both clusters (segments) originate from the same speaker. In the clustering process, the two clusters with the highest cosine distance score are merged together in each iteration. If the maximum value of the similarity for any pair of clusters drops under a given threshold, the stopping condition of the process is met.

Because Eq. (10) can be applied only to a pair of i-vectors, the i-vector representing a certain cluster is simply computed as an *average* of all i-vectors corresponding to the initial segments assigned to the given cluster so far (Franco-Pedroso et al., 2010).

The Linear Discriminant Analysis (LDA) is employed to cope with the nuisance intra-speaker variability. LDA defines an orthogonal projection matrix that maximizes between-class variability and minimizes intra-class variability. In our case, for LDA estimation purpose, a class is formed by all segments of a single speaker in an audio stream, i.e., segments originated by the same speaker but belonging to different streams are assigned to separate classes.

5.4. Diarization output smoothing

The aim of smoothing is to discard either short speech or non-speech intervals that harm the fluency of diarization output, as high granularity of the output is obtrusive to users of the system. Let us remark that in contrast to segmentation and clustering processes, here non-speech events originated by speakers are considered to be parts of speech. First, speech segments shorter than a given number of frames (0.25 s) are discarded. Then, if two speech segments are separated by a non-speech region shorter than a given number of frames (1.0 s), the region is discarded. Depending on whether the adjacent segments are attributed to the same cluster (speaker) or not, they are merged into a single segment, or they are expanded and a change-point is relocated to the middle of the discarded non-speech region.

5.5. Diarization system setup

The diarization system operates with feature vectors formed by 12 Mel-Frequency Cepstral Coefficients (MFCCs). In the first BIC-based clustering pass, we used a BIC penalty weight α of 3.5 and zero value of the clustering stop threshold. For the second clustering stage, feature vectors were mean normalized (within a 3 s long floating window). The universal background GMM with 256 Gaussians was employed for extraction of sufficient statistics. We used 400-dimensional i-vectors and the LDA dimensional reduction to 200.

5.6. Speaker diarization evaluation metrics

Performance of diarization systems is evaluated by the Diarization Error Rate (DER) (NIST, 2009), which can be decomposed as DER = SPKE + FA + MISS, i.e., the speaker error rate, the speech false alarm error rate and the missed speech error rate respectively. The SPKE reflects the amount of speech data that is attributed to a wrong speaker, given the optimum speaker mapping between a system output and a reference diarization. While SPKE is predominantly affected by performance of the speaker clustering module, the FA and MISS evaluate the performance of the speech activity detection module. The NIST scoring tool[3] was employed to compute the metrics in our experiments. A forgiveness collar of 0.25 s (both + and −) was not scored around each boundary.

6. Development of the speaker adaptation approach

Given a speaker diarization output, the state-of-the-art systems for offline transcription of spoken documents employ various SA schemes to improve recognition accuracy. This task is obviously challenging, namely because (a) there is generally no prior information on the speakers occurring in the documents and (b) adaptation has to be performed as unsupervised, using only adaptation data transcripts created by the speech recognizer.

Two main groups of techniques, model-based and feature-based, can be employed for solving this task.

The methods belonging to the former group, e.g., the well-known Maximum A Posteriori (Gauvain and Lee, 1994) (MAP) or Maximum Likelihood Linear Regression (MLLR) (Gales and Woodland, 1996), utilize phonetic transcriptions of adaptation data to update the model parameters. MAP is based on Maximum Likelihood (ML) estimation of new Speaker Dependent (SD) parameters from adaptation utterances. The adapted parameters are then obtained as a linear interpolation (weighted

[3] http://itl.nist.gov/iad/mig/tests/rt/2006-spring/code/md-eval-v21.pl.

sum) between the original speaker independent and new SD parameters. In contrast, MLLR estimates a linear transformation of the original model parameters (usually mean vectors) to create a more appropriate model.

The latter group of feature-based techniques can be used for adaptation or normalization of feature vectors without changing parameters of the model, which can be very large particularly for triphones. Moreover, the methods from this group, e.g., Constrained MLLR (CMLLR) (Gales, 1998) or Vocal Tract Length Normalization (VTLN) (Acero and Stern, 1991; Lee and Rose, 1996), can be employed more easily for building the corresponding normalized acoustic model within Speaker Adaptive Training (SAT) schemes (Anastasakos et al., 1996; Welling et al., 2002).

In contrast to MLLR, the transformation estimated by CMLLR is constrained as the transformation matrix applied to means is the same as the one used for adaptation of covariances. Therefore, the adaptation can also be performed in the feature space; it is only necessary to include the Jacobian of the transformation (i.e., determinant of the transformation matrix) in the likelihood calculation.

The aim of VTLN is to compensate for different lengths of the vocal tracts of individual speakers. This is accomplished by warping the frequency axis during signal parameterization. Similarly as for CMLLR, it is necessary to compute the Jacobian of the warping transformation. Unfortunately, VTLN transformation is typically non-linear and therefore the Jacobian is normally just approximated. To overcome this problem, various implementations of VTLN based on linear transformations have been introduced. These modifications are given in the form of a mathematic equation, e.g., in Acero and Stern (1991), or make use of a training scheme where a set of linear transforms is created in order to approximate the conventional VTLN warping (Kim et al., 2004). Linear VTLN approaches are particularly suitable for the use in transcription tasks where speakers change, because the linear transformation can be switched according to the output from speaker diarization.

Usually, many of the previously mentioned methods are combined to achieve the lowest WER possible. As mentioned in Section 2, this combination can be carried out sequentially or in parallel, based on results obtained in experimental evaluations, and with respect to characteristics of the target task and the ASR module used. However, multiple recognition passes are also associated with high computational demands.

In this work, low RTF is an important factor. Therefore, the aim was to find an approach that would reduce WER significantly but which also had reasonable computational demands. An experimental study was carried out for this purpose, where different outputs from the diarization module were employed for adaptation based on MLLR, MAP and CMLLR complemented with SAT. Linear VTLN approaches were not used as they in effect represent a specially constrained form of CMLLR (see also Povey et al., 2011). Hence in Uebel and Woodland (1999), no significant advantage was reported for the combination of linear VTLN and CMLLR. Therefore, we prefer to make use of GD models, which take into account the difference in vocal tract length between male and female speakers.

Individual performed experiments and all obtained results are detailed in the following sub-sections. The overview and comparison of the most important results reached on the development set is then also presented in Table 9 at the end of Section 6.6.

6.1. Development data

The target archive of Czech Radio contains various types of recordings, particularly representing news programs. However, debates, talk shows and interviews also form an important part of the archive. Hence the development set used within this work has a similar structure. It consists of recordings of daily news (NEWS), talk shows (TS), political debates (DEB) and regional news reports (REP) summarizing the most significant events in one week. An overview of these programs is given in Table 1. They are all of recent origin and their audio quality is good.

Table 1 shows that the chosen programs differ in several characteristics that are important from an automatic speech processing point of view. For example, around 52 different speakers occur on average in one news program from our development set while the recordings of talk shows are composed of utterances belonging only to 3, 4 or 5 speakers (see the third row of Table 1). On the other hand, the debates and talk shows contain a lot of overlapping speech and spontaneous utterances while read-speech is more often present in news programs. The latter difference is important, especially in terms of speech recognition as illustrated by an experiment where just the single decoding pass was performed using the SI model and the 550k lexicon. The results of the experiment presented in the last row of Table 1 show that a significantly lower WER was reached for both types of news programs than for debates and talk shows. The highest WER was obtained for talk shows as speech of interviewed persons was in most cases more spontaneous and less rhetorical (i.e., containing more fillers, repetitions and unfinished sentences) than speech of politicians in the case of political debates.

6.1.1. Performance of speaker diarization

Given the transcripts from the single SI decoding pass, speaker diarization was carried out using the approach

Table 1
The structure of recordings used as the development set.

	NEWS	TS	DEB	REP	Total
# of words	39k	35k	38k	14k	126k
# of programs	10	5	4	7	26
Length (min)	237	258	250	101	846
# speakers per program	52	4	11	24	29
Data (min) per speaker	0.5	14.3	5.5	0.6	1.1
OOV (%)	1.03	1.57	0.77	0.67	0.92
Baseline SI WER (%)	18.04	30.18	22.05	14.73	22.24

Table 2
Results of the speaker diarization module on the development set.

	NEWS	TS	DEB	REP	Total
FA	1.7	0.7	0.2	0.4	0.8
MISS	2.1	1.3	0.7	0.8	1.3
SPKE	13.1	6.0	10.8	8.9	9.7
DER	16.9	8.2	11.6	10.1	11.9
Data (min) per detected speaker	0.4	4.0	2.1	0.5	0.9

detailed in Section 5. All obtained results are summarized in Table 2. They show that the highest DER of 16.9% was reached for daily news as this data has a lot of background noise and is composed of utterances belonging to the highest number of different speakers. In contrast, the speaker clustering module worked particularly well in the case of talk shows, which contain the longest speaker homogeneous segments. The second lowest DER was reached for regional news reports, as these recordings are composed of several tens of shorter reportages with easily distinguishable characteristics of the transmission channels. They also contain less background noise and music than daily news. A good value of DER was also obtained for debates, which are difficult for diarization as they entail a lot of overlapping speech leading to a high value of speaker false alarms.

In terms of speech/non-speech detection, low values of FA and MISS were reached for all programs. This fact proves the good performance of the detection technique we employed, based on utilization of speech transcripts.

6.2. Segment-based adaptation

In order to find out the most suitable adaptation method, a naive approach based on uniform segmentation without the use of any speaker diarization module was performed at first. This allowed us to completely dismiss the speaker diarization module and speed up the adaptation process. Hence, every input recording was just divided into segments of equal duration. Every segment was then transcribed using the SI model and the full 550k lexicon. Next, the obtained phonetic transcripts were employed for global CMLLR-based adaptation using the SI model as initial (prior). As a result of this adaptation step, one transformation matrix was estimated for every segment. During the second decoding pass, the SI model was used along with application of individual transforms to all feature vectors belonging to the corresponding segments.

Results of this experiment are presented in terms of WER in Table 3. The table shows that the uniform segmentation approach was carried out for segment durations of 0.5, 1, 5 and 10 min. The value marked 'whole recording' in the table corresponds to the uniform segmentation experiment considering the entire recording a single segment.

The results show that CMLLR adaptation over the whole recording performed surprisingly well. WER yielded in this scenario was significantly lower than WER yielded in the baseline SI decoding pass. It is also evident that uniform segmentation led to an additional small decrease in WER: the lowest WER was reached for a segment duration of 1 min.

Table 3
Results after the second decoding pass for CMLLR adaptation using uniform segmentation. The total baseline SI WER was 22.24% (see Table 1).

Segment duration (min)	0.5	1	5	10	Whole recording
WER (%)	20.49	20.31	20.35	20.66	20.65

Note that the best value of relative Word Error Rate Reduction (WERR) was obtained for talk shows. The reason is that this data contains the lowest number of speakers per program while the amount of data per speaker is the highest (see Table 1). Therefore, there is the biggest probability that a speaker homogeneous segment is generated by uniform segmentation. The lowest relative improvement over the baseline was obtained for daily news for exactly the opposite reasons.

The next contrast experiment was based on the use of the speaker segmentation module (without speaker clustering). Given the transcript from the first decoding pass, this module was employed to determine single speaker segments. Similar to the previous experiment, global CMLLR-based adaptation was then carried out for every segment. After that, the resulting set of transforms was applied along with the SI model in the second decoding pass. The segments with a too short duration, for which the estimation of the transformation matrix failed, were recognized using only the SI model.

Obtained results are presented in Table 4. Their comparison with those yielded in the former experiments shows that the use of speaker turn detection for adaptation does not lead to better results than the naive segmentation approach. However, this module is still important for the other reason: speaker change-point detection is a preliminary step for clustering in our speaker diarization approach. For the same reasons as in the case of uniform segmentation, the best WERR was also reached for talk shows and the worst for news.

Table 4
Comparison of results after the second decoding pass for segment-based CMLLR adaptation. The segments were determined (a) using uniform segmentation with a duration of 1 min and (b) as speaker-homogeneous based on the output from the speaker segmentation module. The total baseline SI WER was 22.24% (see Table 1).

Program	NEWS	TS	DEB	REP	total
Uniform segmentation for segment duration of 1 min					
WER (%)	17.21	26.54	20.44	13.17	20.31
Rel. WERR over SI (%)	4.6	12.0	7.3	10.6	8.7
Segmentation based on speaker turn detection					
WER (%)	17.56	26.25	20.58	13.36	20.40
Rel. WERR over SI (%)	2.7	13.0	6.7	9.3	8.3

6.3. Cluster-based adaptation

To improve the results obtained so far, several different approaches to cluster-based adaptation were adopted and evaluated. In all instances, the SI decoding pass was first carried out for every input recording. The resulting transcript was then used within the speaker diarization process as a substitute for output of a speech/non-speech detector and to produce a list of admissible change-points at boundaries of transcript elements.

The first experiment employed CMLLR with the SI model as prior, to estimate one transformation matrix for each determined speaker. The SI model and the resulting adaptation matrices were then applied during the second decoding pass similar to what was done in the previous experiment.

Obtained results are presented in Table 5. They demonstrate that cluster-based CMLLR yielded just 3.0% relative WER reduction over segment-based CMLLR (from 20.40% to 19.79%). However, speaker clustering helped particularly in the case of news, where segment-based CMLLR was capable of reducing WER over the SI model by only 2.1% (see the last row of Table 4).

The next goal was to investigate how to further improve the results of cluster-based adaptation. Our motivation stems from the fact that data available for determined speakers across the whole recording exceeds the amount of data in individual speaker homogeneous segments. Hence we are not restricted to techniques suitable for situations dealing with a small amount of available adaptation data, such as in the case of the CMLLR technique with a single global transformation. Therefore at first, CMLLR with four regression classes was employed instead of global CMLLR to allow for more detail and more accurate estimation of the linear transformation.

After that, global MLLR and MLLR with 4 regression classes were employed for adaptation of mean vectors of the SI model. The resulting speaker-specific models were then used for transcription of appropriate speech segments during the second decoding pass. It must also be noted that all models created by MLLR have the same structure of physical states, because they originate from one and the same SI model. Therefore, they can be quickly and easily swapped during decoding.

Finally, global MLLR was followed by MAP to allow for more accurate adaptation of triphones with a higher occurrence in the adaptation data.

Table 5
Detailed results after the second decoding pass using cluster-based global CMLLR adaptation. The total SI WER was 22.24% (see Table 1). Segment-based CMLLR using speaker turn detection yielded WER of 20.40% (see Table 4).

Program	NEWS	TS	DEB	REP	total
WER (%)	16.84	25.90	19.89	12.62	19.79
Rel. WERR over SI (%)	6.6	14.2	9.8	14.3	11
Rel. WERR over Segm. CMLLR (%)	4.1	1.4	3.4	5.5	3.0

Table 6
Total results after the second decoding pass using various approaches for cluster-based adaptation. The total SI WER was 22.24% (see Table 1).

Method	Total WER (%)
Global CMLLR	19.79
CMLLR with four reg. classes	19.93
Global MLLR	20.66
MLLR with four reg. classes	20.28
MLLR + MAP	21.31

Results of this experiment are presented in Table 6. They are somewhat surprising as global CMLLR was not outperformed by any other technique. Only a slightly higher WER was reached by using CMLLR with regression classes while model-based techniques yielded even significantly worse results.

6.4. Two-phase cluster-based adaptation

The positive conclusion of the previous experiment is that cluster-based adaptation using all evaluated adaptation methods leads to WER reduction over the baseline SI model. Hence, the aim of the next experiment was to investigate if it is possible to further improve the results by combining some of these techniques.

This combination was based on a two-phase adaptation approach, which was performed for each speaker as follows: In the first phase, the given SI model, all available feature vectors assigned to the speaker and the corresponding transcriptions were employed to estimate global CMLLR as this technique yielded the lowest WER in the previous experiment. After that, the output transformation matrix was applied to transform all feature vectors of adaptation data. The transformed features and the same phonetic transcriptions as in the first phase were then used for the second phase of adaptation. In this phase, the following three methods were utilized: CMLLR with regression classes and MLLR with and without regression classes.

During the second decoding pass, global CMLLR was applied on all feature vectors belonging to appropriate speech segments at first. The transformed features were then recognized using multiple CMLLR transforms or the appropriately adapted model created by MLLR.

The conducted experiment is summarized in Table 7. The table presents only the total WER values as the obtained WER reductions were similar for all types of programs. The results demonstrate that just the two-phase

Table 7
WER (%) after the second pass using two-phase cluster-based adaptation. One-phase cluster-based global CMLLR yielded WER of 19.79% (see Table 5).

The method in the second phase	CMLLR with four reg. classes	Global MLLR	MLLR with four
WER (%)	20.11	19.71	19.75

adaptation approach based on a combination of global CMLLR and MLLR yielded lower WER than one-phase adaptation. However, this WER reduction was negligible while the computational complexity of the adaptation process was two times higher, because it was necessary to process the adaptation data twice.

6.5. The performance of gender dependent models

The next idea was to take advantage of gender dependent models, since they should generally outperform the SI model. Hence we performed a contrast experiment to determine the performance of GD models at first. Within the experiment, each speech segment was transcribed using the appropriate gender-specific model according to the output from the diarization module. The used GD models were created from the SI model in several additional training iterations on gender-specific data. As the result of the experiment, total WER of 21.68% was obtained. This value proves better performance of GD models as the SI's WER was reduced by 0.56% absolutely.

6.6. The resulting adaptation approach

The resulting adaptation approach takes advantage of all findings from the previous experiments. Therefore, the approach

- relies on the use of GD models
- is cluster-based and employs global CMLLR, as this method yielded better results than CMLLR with regression classes and MLLR (see Section 6.3).
- does not make use of two-phase adaptation due to a negligible additional improvement in accuracy that this technique brings (see Section 6.4).

The scheme of the entire adaptation process is depicted in Fig. 3. It can be described as follows: At first, given the transcripts from the first SI decoding pass, global CMLLR is employed to estimate the transform for every speaker detected by the diarization module. In contrast to previous experiments, the appropriate gender-specific model is used as initial for CMLLR. Both GD models and speaker-specific

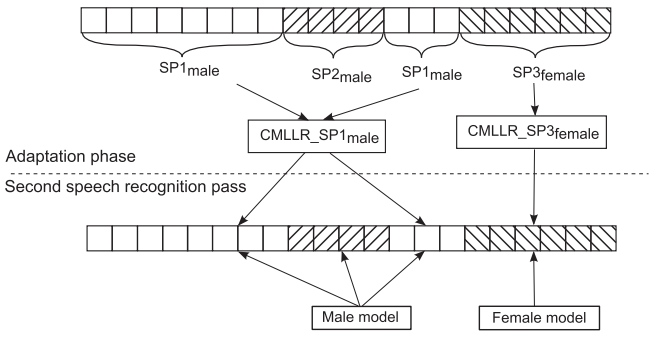


Fig. 3. The resulting cluster-based adaptation scheme uses global CMLLR and GD models created within the SAT framework.

transforms are then used in the second decoding pass, where each transform is applied on all feature vectors belonging to the corresponding speaker. When the amount of adaptation data assigned to the given speaker is below the threshold required for estimation of the transformation matrix, only the appropriate GD model is used for decoding. Moreover, GD models are created within the SAT scheme to further improve recognition accuracy. The results obtained using this approach (with and without SAT) are presented in Table 8.

The upper part of the table shows that the use of GD models as prior for adaptation reduced WER of cluster-based CMLLR from 19.79% to 19.22%. This absolute reduction by 0.57% is nearly the same as the value of 0.56% reached in the previous experiment (see Section 6.5), where GD models were employed instead of the SI model without any adaptation. The lower part of Table 8 demonstrates that the SAT procedure yielded further WER reduction to 18.79%. This is absolute improvement by 0.43%, which is similar to the improvement yielded by GD models.

On average, the resulting approach reduced WER over the SI model by 15.5% relatively. The best values of relative WERR, 20.4% and 19.4%, were reached for recordings of regional news reports and talks shows respectively. The former recordings had the lowest baseline WER of 14.73% while the highest baseline WER of 30.18% was reached for the latter data. This means that the obtained gain in recognition accuracy is not as dependent on the baseline accuracy as we had originally hypothesized. The mentioned impact can rather be attributed to the speaker diarization performance, as the news reports had the second lowest DER of 10.1% and the lowest DER of 8.2% was reached for talk shows. Moreover, the lowest relative WERR of 11.9% was obtained for daily news where the highest DER of 16.9% was yielded (all DER values are presented in Table 2).

In other words, there exists a negative correlation between relative WERR and DER, which is illustrated for all individual programs in Fig. 4. The correlation coefficient of this dependency has a value of -0.39. We also calculated the probability of getting a correlation as large as -0.39 by random chance, when the true correlation is zero. The obtained value of 4.6% proves that the negative correlation is statistically significant.

We also performed an oracle experiment, where the ground truth speaker labels were used instead of the ones produced by the diarization engine. As a result, we obtained just slight additional reduction in WER by 0.32% absolutely on average – like in our previous work (Silovsky et al., 2012). The main reason is that the WERR reached for programs with a low value of DER is on the level of saturation (from speaker adaptation point of view) and cannot be much improved even if speaker labels are produced manually (i.e., with a low number of mistakes).

The last Table 9 presented in this section contains the overview of the most important results reached on the

Table 8
Results after the second pass using the resulting cluster-based adaptation approach with and without application of SAT for GD models building. The total initial SI WER was 22.24% (see Table 1). Global cluster-based CMLLR using the SI model as initial yielded WER of 19.79% (see Table 5).

Program	NEWS	TS	DEB	REP	Total
Global cluster-based CMLLR using GD models as prior					
WER (%)	16.30	25.13	19.44	12.12	19.22
Rel. WERR over cluster-based CMLLR with the SI model (%)	3.2	3.0	2.3	4.0	2.9
Global cluster-based CMLLR using GD models created within SAT					
WER (%)	15.89	24.44	19.22	11.73	18.79
Rel. WERR over cluster-based CMLLR with the SI model (%)	5.7	5.6	3.3	7.0	5.0
Rel. WERR over SI (%)	11.9	19.0	12.8	20.4	15.5

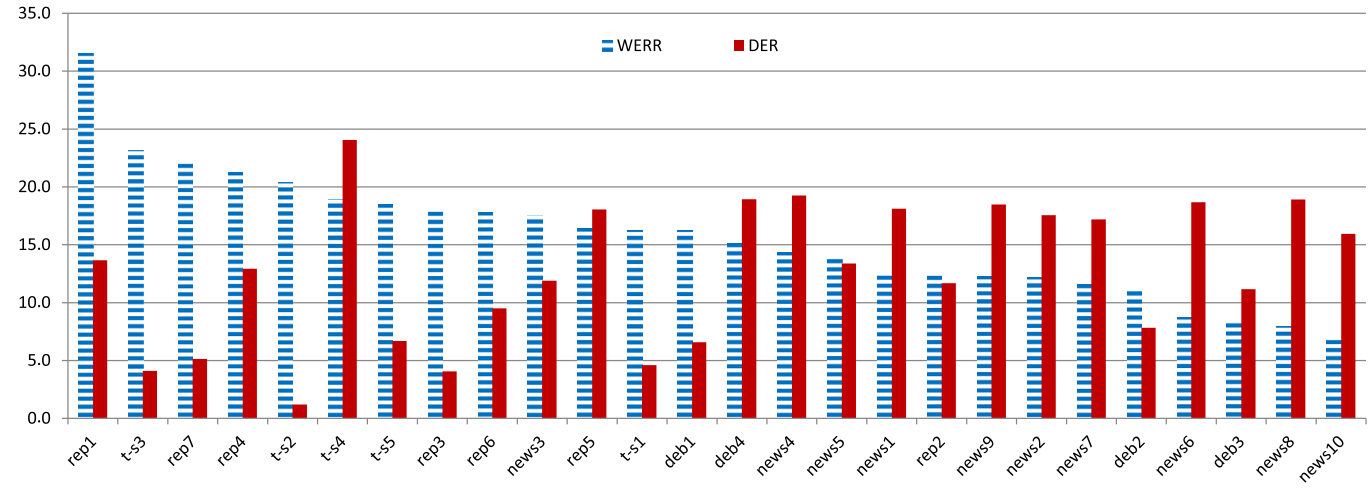


Fig. 4. Obtained values of relative WERR (%) and DER (%) for individual programs from the development set. The results are presented in the descending order according to relative WERR.

Table 9
Overview of the most important results reached on the development set using different adaptation approaches.

Approachesused for recognition (adaptation)	WER (%)
SI model without any adaptation	22.24
Segment-based approaches	
Uniform segmentation (one-minute segm.) + glob. CMLLR	20.31
Speaker Turn Detection (STD) + glob. CMLLR	20.40
STD + GD models without CMLLR	21.68
Cluster-based approaches	
STD + clustering + glob. CMLLR	19.79
STD + clustering + glob. CMLLRfollowed by glob. MLLR	19.71
STD + clustering + glob. CMLLR + GD models as prior	19.22
STD + clustering + glob. CMLLR + SAT GD models	18.79

development set. It shows in one place average values of WER yielded by individual speaker adaptation approaches that were described in the previous sub-sections.

6.7. Reduction of computational demands

Our final efforts were devoted to reducing computational demands of the entire proposed adaptation scheme. The performed analysis showed that the total RTF of the scheme is approximately 1.08 (measured while using one core of a machine with Intel Core i7-2600K@3.4GHz). However, the processes of speaker diarization as well as CMLLR-based adaptation have only small RTFs of around 0.1 and 0.05 respectively. The rest of the computation time is consumed by the first speech decoding pass.

To speed up the decoding in the first pass, we chose the approach based on reduction of the lexicon's size. The reasons are twofold. First, the number of lexical items for decoding can be adjusted by simply changing the appropriate parameter in the decoder. Second, this approach yielded good results in our preliminary study (Cerva et al., 2011), where feature-based adaptation was performed for manually prepared single speaker utterances with a short average length.

In the current work, the lexicon's size in the first pass was limited in the range from 5k to 550k of the most frequent words. The created transcriptions were then employed for adaptation based on the approach described in the previous subsection. All obtained results are summarized in Table 10 and also depicted in Fig. 5 for illustration.

These results demonstrate that the number of lexical items in the first pass can be restricted to the level of 20k

Table 10
Results obtained after using lexicons of a different size in the first decoding pass.

Lexicon's size	5k	10k	20k	40k	50k	100k	300k	550k
The first decoding pass with the SI model								
RTF	0.47	0.50	0.53	0.58	0.59	0.69	0.87	0.93
WER (%)	61.50	49.33	38.90	31.10	29.31	24.95	23.34	22.24
CER (%)	24.04	18.88	15.22	12.53	12.03	10.15	9.48	7.59
DER (%)	11.9	11.5	11.6	11.4	11.6	11.3	11.4	11.9
The second pass using the 550k lexicon and the resulting adaptation approach								
WER (%)	19.51	19.21	**18.85**	18.82	18.81	18.79	18.80	18.79

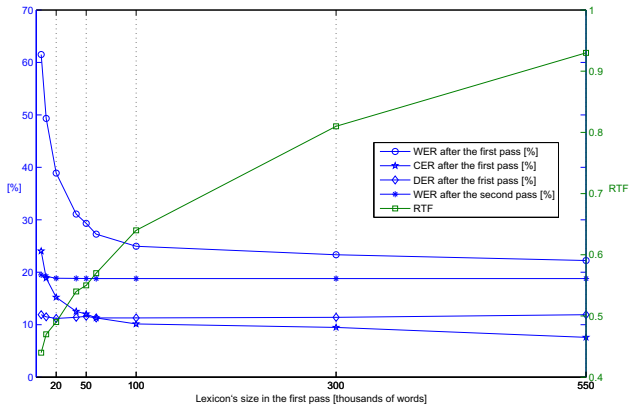


Fig. 5. Dependency of WER, DER, CER and RTF on the size of a lexicon employed in the first decoding pass.

words. The recognition system then operates at 0.53 RTF and although a very high WER of 38.90% is yielded, the generated transcriptions are still accurate enough for speaker diarization and the estimation of the CMLLR transform: the reached 11.6% DER was slightly better than DER obtained with the full lexicon and the yielded WER of 18.85% after the second pass was slightly worse.

The reason for these results is the inflective character of the Czech language, which has many word forms that often differ in only one or several characters in the prefix, suffix or word ending. Hence, during recognition with a restricted lexicon, every missing word is usually substituted by a similar word or by several short words that form together a similar sequence of phonemes. This fact is illustrated in the fourth row of Table 10, where Character Error Rate (CER) is presented. For example, the absolute increase in WER by 16.5% (from 22.24% to 38.9%) for the 20k lexicon corresponds to the absolute increase in CER just by 7.63% (from 7.59% to 15.52%). It is also evident that the limitation of the lexicon's size under the level of 20k words does not bring any further significant RTF reduction, and that a decrease in WER after the second pass was observed in this case.

It should also be noted that in Cerva et al. (2011), a similar decrease in WER was observed when the lexicon's size was limited to 40k words, which is a higher number. The reason is that a smaller amount of adaptation data was available (12.7 s on average) in this study, because the process of adaptation was carried out for individual single-speaker utterances and no speaker clustering module was employed. In the current work, the average amount of data per speaker is much higher: about 54 s.

7. Conclusions

The current paper describes a speaker-adaptive speech recognition scheme that can be used to improve transcription accuracy of spoken documents in situations when the employed ASR system has to operate at a reasonable RTF. Therefore, the proposed framework makes use of only two recognition passes and takes advantage of speech transcripts not only for adaptation, which is usual, but also in the speaker diarization process. We showed that the preliminary transcripts can be obtained using a lexicon with a limited number of items to significantly speed up adaptation without increasing DER and losing recognition accuracy after the second decoding pass.

Given the output from the diarization module, an experimental study was performed for various radio programs in order to find out the most suitable adaptation approach. The resulting adaptation method is cluster-based and employs global CMLLR transform, as this technique yielded the lowest WER. The approach does not make use of VTLN; it rather relies on GD models. These models are used in the second decoding pass and as prior for CMLLR instead of the SI model. Note that the relative decrease in WER of 2.9%, which was obtained for cluster-based CMLLR from the use of GD models, is similar to the additional relative WER reduction of 3.3%, which was yielded by combining CMLLR with VTLN in Cerva et al. (2011). The obtained results also show that there is a significant correlation between diarization accuracy and the yielded WER reduction.

Acknowledgement

This work was supported by the Czech Science Foundation (project no. P103/11/P499) and by the Student Grant Scheme (SGS) at the Technical University of Liberec.

References

Acero, A., Stern, R., 1991. Robust speech recognition by normalization of the acoustic space. In: Acoustics, Speech, and Signal Processing, 1991.

ICASSP-91, 1991 International Conference on, vol. 2, ISSN 1520-6149, pp. 893–896.

Anastasakos, T., Mcdonough, J., Schwartz, R., Makhoul, J., 1996. A Compact Model for Speaker-Adaptive Training. In: Proc. ICSLP, pp. 1137–1140.

Anguera, X., Wooters, C., Pardo, J., 2006. Robust speaker diarization for meetings: ICSI RT06s evaluation system, in: Interspeech'06. Pittsburgh, PA, USA.

Breslin, C., Chin, K.K., Gales, M.J.F., Knill, K., 2011. Integrated online speaker clustering and adaptation. In: INTERSPEECH, pp. 1085–1088.

Byrne, W., Doermann, D., Franz, M., Member, S., Gustman, S., Soergel, D., Ward, T., jing Zhu, W., 2004. Automatic Recognition of Spontaneous Speech for Access to Multilingual Oral History Archives. IEEE Transactions on Speech and Audio Processing 12, 420–435.

Cerva, P., Palecek, K., Silovský, J., Nouza, J., 2011. Using unsupervised feature-based speaker adaptation for improved transcription of spoken archives. In: Proceedings of Interspeech 2011, International Speech Communication Association, pp. 2565–2568.

Chen, S., Gopalakrishnan, P., 1998. Speaker, environment and channel change detection and clustering via the Bayesian information criterion. In: Proceedings 1998 DARPA Broadcast News Transcription and Understanding Workshop, pp. 127–132.

Chu, S.M., Kuo, H.-K., Mangu, L., Liu, Y., Qin, Y., Shi, Q., Zhang, S.L., Aronowitz, H., 2008. Recent advances in the IBM GALE Mandarin transcription system. In: Acoustics Speech and Signal Processing (ICASSP), 2008 IEEE International Conference on, pp. 4329–4332.

Chu, S., Povey, D., Kuo, H.-K., Mangu, L., Zhang, S., Shi, Q., Qin, Y., 2010. The 2009 IBM GALE Mandarin broadcast transcription system. In: Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on, ISSN 1520-6149, pp. 4374–4377.

Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., Ouellet, P., 2011. Front-End Factor Analysis for Speaker Verification, Audio, Speech, and Language Processing, IEEE Transactions on 19 (4) (2011) 788–798, ISSN 1558-7916, doi:10.1109/TASL.2010.2064307.

Fiscus, J.G., 1997. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). In: Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding, pp. 347–354.

Franco-Pedroso, J., Lopez-Moreno, I., Toledano, D., Gonzalez-Rodriguez, J., 2010. ATVS-UAM System Description for the Audio Segmentation and Speaker Diarization Albayzin 2010 Evaluation. FALA 2010, 415–417.

Gales, M., 1998. Maximum likelihood linear transformations for HMM-based speech recognition. Computer Speech and Language 12, 75–98.

Gales, M., Woodland, P., 1996. Mean and variance adaptation within the MLLR framework. Computer Speech and Language 10, 249–264.

Gauvain, J., Lee, C., 1994. Maximum a posteriori estimation for multivariate gaussian mixture observations of Markov chains. IEEE Transactions on Speech and Audio Processing 2, 291–298.

Gauvain, J.-L., Lamel, L., Adda, G., 1998. Partitioning and transcription of broadcast news data. In: ICSLP'98, pp. 1335–1338.

Glass, J., Hazen, T.J., Cyphers, S., Malioutov, I., Huynh, D., Barzilay, R., 2007. Recent progress in the MIT spoken lecture processing project. In: Proceedings of InterSpeech 2007, ISCA, pp. 2553–2556.

Hain, T., Burget, L., Dines, J., Garner, P., Grezl, F., Hannani, A., Huijbregts, M., Karafiat, M., Lincoln, M., Wan, V., 2012. Transcribing Meetings With the AMIDA Systems Audio Speech and Language Processing. IEEE Transactions on 20 (2), 486–498, ISSN 1558-7916.

Hansen, J.H.L., Huang, R., Zhou, B., Seadle, M., Deller, J.R., Gurijala, A.R., Kurimo, M., Angkititrakul, P., 2005. SpeechFind: Advances in Spoken Document Retrieval for a National Gallery of the Spoken Word Speech and Audio Processing. IEEE Transactions on 13 (5), 712–730, ISSN 1063-6676.

Kenny, P., Boulianne, G., Dumouchel, P., 2008. Eigenvoice modeling with sparse training data. IEEE Trans. Processing 13, 345–354.

Kim, D.Y., Umesh, S., Gales, M.J.F., Hain, T., Woodland, P.C., 2004. Using VTLN for broadcast news transcription. In: Proceedings of 2004 International Conference Spoken Lang. Process, Jeju Island, South Korea

Kneser, R., Ney, H., 1985. Improved backing-off for M-gram language modeling. In: IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), vol. 1, 1995, pp. 181–184.

Lee, L., Rose, R., 1996. Speaker normalization using efficient frequency warping procedures. In: Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings, 1996 IEEE International Conference on, vol. 1, ISSN 1520-6149, pp. 353–356.

Liu, D., Kiecza, D., Srivastava, A., Kubala, F., 2005. Online speaker adaptation and tracking for real-time speech recognition. In: INTERSPEECH, ISCA, pp. 281–284.

Mangu, L., Brill, E., Stolcke, A., 2000. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. Computer Speech & Language 14 (4), 373–400.

Matsoukas, S., Gauvain, J.-L., Adda, G., Colthurst, T., Kao, C.-L., Kimball, O., Lamel, L., Lefevre, F., Ma, J., Makhoul, J., Nguyen, L., Prasad, R., Schwartz, R., Schwenk, H., Xiang, B., 2006. Advances in transcription of broadcast news and conversational telephone speech within the combined EARS BBN/LIMSI system Audio Speech and Language Processing. IEEE Transactions on 14 (5), 1541–1556, ISSN 1558-7916.

Moattar, M., Homayounpour, M., 2012. A review on speaker diarization systems and approaches, Speech Communication 54 (10) (2012) 1065–1103, ISSN 0167-6393.

Ng, T., Zhang, B., Nguyen, L., Matsoukas, S., Zhou, X., Mesgarani, N., Veselý, K., Matějka, P., 2012. Developing a speech activity detection system for the DARPA RATS program. In: Proceedings of Interspeech 2012, International Speech Communication Association, pp. 1967–1970.

NIST, 2009. The 2009 (RT-09) Rich Transcription Meeting Recognition Evaluation Plan.

Nouza, J., Zdansky, J., Cerva, P., Kolorenc, J., 2006. Continual On-line monitoring of Czech spoken broadcast programs. Proceedings of INTERSPEECH 4 (1650–1653), 2006.

Nouza, J., Zdansky, J., Cerva, P., 2010. System for automatic collection, annotation and indexing of Czech broadcast speech with full-text search. In: 15th IEEE MELECON Conference, Malta, pp. 202–205.

Nouza, J., Blavka, K., Bohac, M., Cerva, P., Zdansky, J., Silovsky, J., Prazak, J., 2012. Voice technology to enable sophisticated access to historical audio archive of the Czech radio. In: Multimedia for Cultural Heritage, vol. 247 of Communications in Computer and Information Science, Springer, Berlin, Heidelberg, pp. 27–38.

Ordelman, R., de Jong, F., Heeren, W., 2006. Exploration of audiovisual heritage using audio indexing technology. In: Brewka, G., Coradeschi, S., Perini, A., Traverso, P. (Eds.), Proceedings of the 17th European Conference on Artificial Intelligence (ECAI2006). IOS Press, Amsterdam.

Povey, D., Zweig, G., Acero, A., 2011. Speaker adaptation with an Exponential Transform. In: ASRU, pp. 158–163.

Shinoda, K., 2005. Speaker adaptation techniques for speech recognition using probabilistic models, Electronics and Communications in Japan (Part III: Fundamental Electronic Science) 88 (12) (2005) 25–42, ISSN 1520-6440.

Silovsky, J., Prazak, J., 2012. Speaker Diarization of Broadcast Streams using Two-stage Clustering based on I-vectors and Cosine Distance Scoring. In: ICASSP 2012, Kyoto, Japan, pp. 4193–4196.

Silovsky, J., Cerva, P., Zdansky, J., Nouza, J., 2012. Study on Integration of Speaker Diarization with Speaker Adaptive Speech Recognition for Broadcast Transcription, in: Proceedings of Interspeech 2012, International Speech Communication Association, , pp. 478–481.

Stolcke, A., Anguera, X., Boakye, K., Çetin, O., Janin, A., Magimai-Doss, M., Wooters, C., Zheng, J., 2008. Multimodal Technologies for

Perception of Humans, chap. The SRI-ICSI Spring 2007 Meeting and Lecture Recognition System, Springer-Verlag, Berlin, Heidelberg, ISBN 978-3-540-68584-5, pp. 450–463.

Uebel, L.F., Woodland, P.C., 1999. An investigation into vocal tract length normalisation. In: Sixth European Conference on Speech Communication and Technology, EUROSPEECH 1999, Budapest, Hungary, September 5–9, 1999, ISCA, 1999.

Welling, L., Ney, H., Kanthak, S., Vi, L.F.I., 2002. Speaker Adaptive Modeling by Vocal Tract Normalization. IEEE Transactions on Speech and Audio Processing 10, 415–426.

Zdansky, J., 2006. BINSEG: An efficient speaker-based segmentation technique. In: INTERSPEECH 2006, Pittsburgh, PA, USA, pp. 2182–2185.

4 Dealing with non-speech events

4.1 Speech activity detection in streamed data

The first two articles included in this chapter deal with the SAD task. They were published within two consecutive TAČR projects dealing partially with 24/7 transcription of TV/R streams. Both of these papers are thus devoted to the frame-wise operation mode. Their main author is my former PhD student Lukáš Matějů. The other author of the second article, František Kynych, is my current PhD student.

The first ICASSP paper [23] proposes an on-line SAD approach that utilizes binary DNN-based classifier; its output is smoothed by a WFST-based decoder (see also Sec. 2.2.2). The DNN is trained on 7 hours of various non-speech events and/or noises, 30 hours of music recordings and 30 hours of clean speech utterances belonging to several Slavic languages and English. Moreover, a data-set of mixtures comprising speech and non-speech signals is also employed to further increase the detection accuracy. This additional data is created artificially as follows: At first, a larger set of non-speech recordings of a total length of 100 hours is prepared. After that, every speech recording is mixed with a randomly selected non-speech recording from the prepared set. Within this mixing, the volume of every non-speech recording is increased or decreased to match the desired level of SNR (which is also selected randomly from an interval between -30 dB and 50 dB). The decoder employs a context-based transduction model, i.e., both speech and non-speech events are modeled using sequences of three states.

The data used for development and evaluation consists of 6 hours of TV and radio recordings in several Slavic languages, e.g., Czech, Slovak, Polish or Russian. It contains not only clean speech segments but also segments with music, jingles and/or advertisements. In this data, our approach is capable of detecting speech/non-speech change points with the F-value of 52.7%. At the same time, the achieved frame error rate (FER) is 2.4%. Given these results on the development set, the experimental evaluation is also performed using standardized QUT-NOISE-TIMIT data-set for SAD [24]. This approach yields state-of-the-art results in both low and medium noise conditions while staying competitive under high noise conditions.

The method is also further tested from the ASR point of view by using two sets of Czech broadcasts data. The first one contains 4 hours of recordings from a Czech live news TV channel. Approximately 60% of the relevant content is speech. The second set is compiled from 8 hours of broadcast of a Czech local radio station, from which speech only makes up 10%.

On these data-sets, the SAD module reduces WER from 12.7% to 12.4% and from 17.9% to 14%, respectively. The reason is that it limits the insertions coming from the non-speech parts and hardly omits any speech parts. At the same time, the SAD module operates with low latency around 2s and reduces the computational demands of the target transcription system significantly: the real-time factor of the ASR module is around 1.0, while the SAD module operates with RTF of 0.1.

The second Interspeech paper [25] utilizes the x-vectors [26] as input features for SAD. The motivation for this work stems from the fact, that in real applications utilizing speaker diarization, these two systems must be employed over different features (one for SAD and one for SD). This fact increases computation demands and the complexity of the whole speech processing pipeline, which is undesirable. This is namely true in systems designed for processing of streamed data. In our case, an example of such a system is the platform for 24/7 monitoring of various TV and radio streams (see also Sec. 6.1). In this type of application, the SAD module forms the first block of the data-processing pipeline and its output, which consists of the speech segments, is used for a) speaker diarization (followed by speaker recognition) and b) speech transcription.

Within our method, the x-vectors are extracted by using a vectorized feed-forward sequential memory network (FSMN) [27] and form an input into a computationally undemanding binary classifier (with a sole hidden layer), whose output is again smoothed by a WFST-based decoder. The employed FSMN architecture allows us to model long-time dependencies in the input signal similar to recurrent networks. However, its main advantage is that it eliminates the recursion by adding several memory blocks with trainable weight coefficients into each layer of a standard feed-forward fully connected (FC) DNN. The memory blocks use a tapped-delay line structure to encode the long context information into a fixed-size representation. That means that the FSMN layers are built on top of the context of the earlier layers, the final context is thus a sum of the partial ones.

We train the FSMN extractor by using Voxceleb2 [28], "train-360-clean" part of LibriSpeech [29] and 121 hours of clean Czech microphone recordings belonging to 922 speakers. Moreover, the set of target speakers is also extended by the noise class defined over the train part of the CHiME-4 dataset [30]. The binary classifier is trained over the x-vectors, that are extracted from the data-set consisting of 30 hours of clean speech, 30 hours of music and 30 hours of artificially mixed speech and music/noise recordings according to randomly chosen SNR (the same as in the previous article). All these recordings are concatenated in a random order to contain speech/non-speech transitions.

In comparison with the previous SAD module, the described approach yields slightly better results on the same development set: F-value of 59.8% and FER of 2.2% are achieved. It also gets similar performance on broadcast data and similar results on the QUT-NOISE-TIMIT data-set. However, the main advantage of this method in comparison with the DNN-based approach is in the fact that the x-vectors used for SAD can be employed simply also for speaker diarization or recognition in the next stages of the data-processing pipeline.

4.2 Recognition of speech with background music

The next two articles, published at ICASSP conferences, address the robustness of ASR systems with respect to background music. My contribution to both of them is the multi-condition training of the acoustic model.

The first work [31] investigates this type of training and autoencoders. Three different multi-style models are trained based on piano and electronic music in the background of the training speech. The training data for each of these models is prepared as follows: at first, N desired SNR levels are selected. Subsequently, the speech corpus (consisting of 132 hours of clean speech) is split into $N+1$ parts and the first part is left undistorted. The corresponding music scaled to the predefined average SNR level is added to all other parts. The average SNR is computed per file of speech recordings, which usually corresponds to about two sentences.

The second approach relies on the removal of background music using a denoising autoencoder. Two types of auto-encoders are considered: a fully connected autoencoder (FAE) and a convolutional network autoencoder (CAE). Each of them accepts a sequence of 11 consecutive distorted feature vectors (each feature vector consists of 39 filter bank coefficients) at their input layer and produce an estimate of clean speech features on the output. During the training stage, the autoencoders require pairs of undistorted and corrupted utterances (these are generated artificially as described in the previous paragraph). Their training minimizes the mean square distance between the distorted input and the clean target. This criterion is sensitive to scaling, thus we normalize both the training and test data (each feature separately) to zero mean and unit variance. The same normalization values are utilized later in the test phase. Our FCAE is constituted of three hidden layers, with 1024 neurons in each layer. The CNAE topology differs in two aspects: 1) the input layer; and 2) the substitution of the first hidden layer of the FCAE by two convolutional layers.

The experimental evaluation is performed on the artificially generated data-set at first. The set's duration is 2.45 hours; it consists of dictated texts, recorded in a silent environment via a close-talk microphone. We add to the speech various piano music tracks with several SNR levels. The results obtained on this data show that both studied techniques are able to compensate for the performance decrease (caused by interfering music) encountered by a single-style baseline model. The accuracy achieved by both techniques is comparable to matched train-test conditions and simpler background music. However, the multi-condition models exhibit superior accuracy for mismatched training-test scenarios (with unseen music genre and/or SNR level) and for more complex background music.

The second test set's duration is 17.5 minutes; it comprises real-world speech recordings with music in the background. These recordings come from several summaries given at the beginning of a news program in a radio. A track of electronic music is present in their background. We estimate their average SNR level at about 10 dB. On this data, the multi-condition training as well as the CAE is capable to reduce the baseline WER of 16.3% to 13.9%, while FAE yields slightly lower WER reduction to 14.2%. Finally, it should also be noted that the important advantage

of all the approaches studied is that they do not deteriorate the accuracy under scenarios with clean speech (the decrease is about 3% only).

The second article [32] follows up on the previous work. It investigates the suitability of the above-mentioned techniques under a scenario in which a very small amount of labeled training speech is available (with a duration of about 1 hour). This problem can be encountered, e.g., when building an ASR system for a new, in particular under-resourced, language. Since speech labeling is costly and time-consuming, we also investigate the possibility of improving the performance using a larger amount of unannotated speech data. We compare the performance of these under-resourced AMs to models trained using a large amount of labeled speech.

Next, we extend the portfolio of the investigated techniques to robust ASR. First, we employ multi-condition training of convolutional acoustic models (MCT-CAM). The goal is to take into account the advantages of convolutional topology as reported in [33]. Second, we utilize joint training of the acoustic model and convolutional autoencoder (JMCT) as proposed in [34]. This approach fine-tunes both the acoustic model and the feature enhancement by CAE: it explois the information about the senone classification instead of optimizing just the squared error as in the conventional training of CAE.

Finally, we also carry out a more detailed analysis of the autoencoder performance with respect to its topology than in previous article. There we found that the performance of FAE is comparable to the performance of CAE, assuming that both networks have a comparable number of hidden units. This, however, bestows the CAE with a lower number of free parameters. In this work, we show that CAE yields better results than the FAE when broader convolutional layers and/or deep networks are employed.

The performance of all these methods is again evaluated on artificial mixtures of speech and music as well as using real-world radio shows. The obtained results lead to four main conclusions which hold regardless of the amount of the training data used: First, all investigated approaches are able to improve the results against the baseline model. Second, the CAE is more suitable for noisy feature enhancement than FAE. Third, when multi-condition training is employed, the convolutional AM outperforms the feed-forward fully connected architecture. Finally, best results are achieved using the joint training of the autoencoder and AM. This holds even when comparing MCT-CAM and JMCT, which have similar topologies and sizes. This means that a pre-trained CAE is suitable as the initial layers of the final acoustic model when it is fine-tuned along with the weights of the AM.

The experiments performed using models trained on the small data-set further show that models trained using a smaller amount of data exhibit lesser accuracy and are less robust with respect to background music. An additional amount of unannotated data can considerably improve the performance of any autoencoder type. Moreover, it can also considerably improve the performance of JMCT-based systems.

4.3 Reprints

[23] L. Mateju, P. Cerva, J. Zdansky and J. Malek. "Speech Activity Detection in Online Broadcast Transcription Using Deep Neural Networks and Weighted Finite State Transducers". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*, pp. 5460–5464.

[25] L. Mateju, F. Kynych, P. Cerva, J. Zdansky and J. Malek. "Using X-vectors for Speech Activity Detection in Broadcast Streams". In: *Proceedings of Annual Conference of the International Speech Communication Association, INTERSPEECH, Brno, Czech Republic, 2021*, Accepted to.

[31] J. Malek, J. Zdansky and P. Cerva. "Robust Automatic Recognition of Speech with background music". In: *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018*, pp. 5210–5214.

[32] J. Malek, J. Zdansky and P. Cerva. "Robust Recognition of Speech with Background Music in Acoustically Under-Resourced Scenarios". In: *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018*, pp. 5624–5628.

SPEECH ACTIVITY DETECTION IN ONLINE BROADCAST TRANSCRIPTION USING DEEP NEURAL NETWORKS AND WEIGHTED FINITE STATE TRANSDUCERS

Lukas Mateju, Petr Cerva, Jindrich Zdansky and Jiri Malek

Technical University of Liberec,
Faculty of Mechatronics, Informatics, and Interdisciplinary Studies,
Studentska 2, 461 17 Liberec, Czech Republic

ABSTRACT

In this paper, a new approach to online Speech Activity Detection (SAD) is proposed. This approach is designed for the use in a system that carries out 24/7 transcription of radio/TV broadcasts containing a large amount of non-speech segments, such as advertisements or music. To improve the robustness of detection, we adopt Deep Neural Networks (DNNs) trained on artificially created mixtures of speech and non-speech signals at desired levels of signal-to-noise ratio (SNR). An integral part of our approach is an online decoder based on Weighted Finite State Transducers (WFSTs); this decoder smooths the output from DNN. The employed transduction model is context-based, i.e., both speech and non-speech events are modeled using sequences of states. The presented experimental results show that our approach yields state-of-the-art results on standardized QUT-NOISE-TIMIT data set for SAD and, at the same time, it is capable of a) operating with low latency and b) reducing the computational demands and error rate of the target transcription system.

Index Terms— deep neural networks, speech activity detection, weighted finite state transducers, speech recognition

1. INTRODUCTION

Speech activity detection is a problem of identifying both speech and non-speech segments in a sound recording. Over the years, various SAD approaches have been proposed and an SAD module has usually formed an integral component of a signal pre-processing algorithm in a wide range of tasks including, e.g., speech enhancement, speaker identification and, of course, speech transcription. Most of the existing SAD approaches are carried out in two subsequent stages: feature extraction, and speech/non-speech classification.

In the former phase, the classic approaches for feature extraction utilize energy [1], zero crossing rate [2] or auto-correlation function [3]. The family of more complex features, which have also been successfully applied, include MFCCs [4, 5], multi-resolution cochleagram features [6], multi-band long-term signal variability features [7] or channel bottleneck features [8]. Note that in [9], features based on the use of Deep Belief Networks (DBN) have also been proposed. In practice, various combinations of individual features are usually used to achieve the best possible results.

In the latter phase, various classification algorithms can be used, such as Support Vector Machines (SVM) [10] or Gaussian Mixture Models (GMMs) [11, 12]. In recent years, various DNN architectures started to be employed more and more frequently including fully connected feed-forward DNNs [5], Convolutional Neural Networks (CNNs) [13] or Recurrent Neural Networks (RNNs) [14, 15]. More complex approaches such as jointly trained DNNs [16] or boosted DNNs [6] have also been proposed. Moreover, in [17] a combination of DNN and CNN is used. The output from a given classifier can also be smoothed to further improve the accuracy of the detection. Recently, various techniques such as the Viterbi decoder [5, 18] or WFSTs [19] have been applied for this purpose.

Most of the aforementioned works aim primarily at offline applications, because applying SAD in an online environment brings further restrictions on the system, such as low computational demands and latency. The approaches developed namely for the online task include, for example, conditional random fields [18] or accurate endpointing with expected pause duration [20]. Another approach in [21] utilizes short-term features.

The goal of our efforts was to develop a SAD approach suitable for a system that is deployed for online 24/7 transcription of more than 80 TV and radio stations in several Slavic languages. From a speech recognition point of view, this type of input data is specific by containing a large amount of non-speech parts such as songs or advertisements. This applies namely to some local radio stations, whose broadcasts may contain only a few percent of speech segments. In this case, the utilization of an SAD module should reduce the computation demands on the transcription system dramatically.

An SAD module suitable for our target task should a) operate at a low level of Real-Time Factor (RTF), b) have a low latency, and c) reduce the Word Error Rate (WER) of transcription. To meet all these requirements at once, a new approach is proposed in the present paper. It adopts a DNN classifier that is trained on a data set created by mixing clean speech utterances with non-speech recordings at various desired levels of SNR. The output from DNN is then smoothed using a decoder based on WFSTs. To ensure high quality and accuracy of the detection, the employed transduction model is context-based, i.e., both speech and non-speech events are modeled as sequences of three consecutive states.

2. METRICS USED FOR EVALUATION

In this paper, three different overall accuracy metrics were used for evaluation including Frame Error Rate (FER), Miss Rate (MR) and False Alarm Rate (FAR) [5].

Moreover, the F-measure was utilized to evaluate the quality of the change-point detection between speech/non-speech events given the alignment between detected and reference boundaries [22]. Given the correctly detected boundaries (hits), it is also possible to calculate an error value for each hit (in seconds) and sort all the hits according to these values in ascending order. In this paper, the measure $\delta_{2/3}$ is utilized, which expresses (in seconds) the maximal error of the alignment for first two-thirds of the sorted (best) hits.

3. THE PROPOSED SAD APPROACH

The SAD approach presented in this paper was developed in a series of experiments described in the following subsections.

3.1. Data Used for Development

The data used for development and evaluation consisted of 6 hours of TV and radio recordings in several Slavic languages, e.g., Czech, Slovak, Polish or Russian. It contained not only clean speech segments but also segments with music, jingles and/or advertisements. Annotation of this data was created within two consecutive phases: speech/non-speech labels were produced automatically using the baseline DNN-based SAD approach (see the next section) at first, and then corrected by human annotators. In total, 70% of all frames were marked as containing speech.

3.2. Baseline DNN-Based Approach

The baseline approach employed a deep neural network with a binary output (i.e., without any smoothing) which was trained using the torch library[1]. The data for DNN training was composed of 7 hours of various non-speech events and/or noises, 30 hours of music recordings and 30 hours of clean speech utterances belonging to several Slavic languages and English.

The DNN had 5 hidden layers, each consisting of 128 neurons. The ReLU activation function and mini-batches of size 1024 were used within 10 epochs of training. The learning rate was set to 0.08. 39-dimensional log filter banks were used as features. The input vector for DNN had a length of 51 and was formed by concatenating 25 previous frames, the current frame, and 25 following frames. Local normalization was performed within one-second windows.

The accuracy of the baseline approach is summarized in Table 1 (see its first row). It is evident that it missed approximately 4% of speech segments. This fact affects the accuracy of the speech transcription system negatively, as the segments incorrectly marked as non-speech are not transcribed. Another problem of the baseline detector is the time precision of the change-point detection: the achieved value of $\delta_{2/3}$ is 0.42 s. This is also due to the fact that it is sometimes hard even for human annotators to determine the exact frame where a state change occurs. The baseline detector also produced a high number of false non-speech segments with a very short duration of one or two frames.

3.3. Smoothing the Output from DNN

As mentioned in the previous section, the baseline detector classified every input frame independently. On the other hand, every speech or non-speech segment usually lasts for at least several frames. Therefore, our next efforts were focused on smoothing the output from DNN. For this purpose, weighted finite state transducers were utilized using the OpenFst library[2].

The resulting scheme consists of two transducers (see Fig. 1). The first models the input signal. The other one is the transduction model and represents the smoothing algorithm. It consists of three states. The first state, denoted by 0, is the initial state. The transitions between states 1 and 2 emit the speech/non-speech labels and are penalized by penalty factors P1 a P2, respectively. Their values (500 and 500) were determined in several experiments not presented in this paper. Note that these values were tuned on different data set.

[1] http://torch.ch

[2] http://www.openfst.org/twiki/bin/view/FST/WebHome

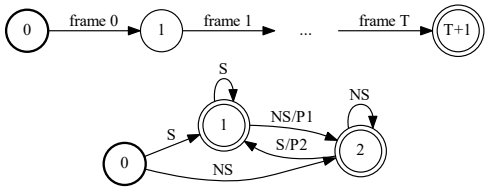


Fig. 1. The transducers representing the input signal (upper) and the basic smoothing model without any context (lower).

Given the two described transducers, the decoding process is performed using on-the-fly composition of the transduction and the input model of unknown size. This is possible since the input is considered to be a linear-topology, unweighted, epsilon-free acceptor. After each composition step, the shortest-path (considering tropical semi-ring) determined in the resulting model is compared with all other alternative hypotheses. When a common path is found among these hypotheses (i.e., with the same output label), the corresponding concatenated output labels are marked as the final fixed output. Since the rest of the best path is not known with certainty, it is denoted as a temporary output (i.e., it can be further refined).

The results obtained with the aid of the DNN-based approach with smoothing are summarized in the second row of Table 1. They show an overall significant boost in accuracy. For example, MR was reduced from 3.7% to 2.2% and the value of $\delta_{2/3}$ from 0.42 s to 0.27 s.

3.4. Using Artificial Training Data

The level of MR yielded so far, i.e., around 2%, still leads to a small increase in WER of a transcription system (e.g., from 13% to 14%), as the speech frames incorrectly classified as non-speech are omitted from transcription. The analysis we performed showed that the detector specifically misclassified the speech segments with background noise. The reason for this behavior is that the speech data used for DNN training so far were recorded in clean conditions (they served originally for training of an acoustic model for speech recognition systems).

Hence in the next step, the goal was to employ training data containing non-speech events, such as music or jingles in the background. The lack of such annotated data forced us to create an artificial source by mixing 30 hours of clean speech with non-speech recordings. For this purpose, a larger set of non-speech recordings of a total length of 100 hours was prepared first. After that, every speech recording was mixed with a randomly selected non-speech recording from the prepared set. Note that every non-speech recording used for mixing had to have the same or longer duration than the given input speech recording (the selected non-speech recording was trimmed to match the length of the speech recording) and its volume was increased or decreased to match the desired level of SNR (which was also selected randomly from an interval between -30 dB and 50 dB).

The labeling of this artificial data was carried out automatically: when SNR of the recording was higher than a defined threshold of 0 dB, the recording was marked as containing speech. In the opposite case, the recording was labeled as non-speech.

The results after using only these 30 hours of mixed training data are shown in the third row of Table 1. It is evident that this approach led to an increase in F-measure and a significant reduction in MR from 2.2% to 0.3%. Unfortunately, these improvements are

Table 1. Summarized results of individual SAD approaches described in Sect. 3.

Approach	FER	MR	FAR	F-measure	$\delta_{2/3}$
Baseline DNN-based	4.7%	3.7%	7.1%	0.3%	0.42 s
+ Basic smoothing	2.9%	2.2%	4.7%	28.5%	0.27 s
+ Artificial training data with noise	3.1%	0.3%	10.1%	41.3%	0.34 s
Modified artificial training data + Context-based smoothing	2.4%	0.5%	7.2%	52.7%	0.26 s

all accompanied by an increase in FAR and, even more importantly, an increase in $\delta_{2/3}$ from 0.27 s to 0.34 s. This negative fact motivated us to further improve the smoothing algorithm.

3.5. Improved Context-Based Smoothing

The scheme of the improved smoothing transducer that utilizes context is depicted in Fig. 2. In this case, both the speech and non-speech events are represented as sequences of three states, where the first and third states (the outer states) model the context. Similarly to smoothing without any context, the penalties are defined just for transitions between the speech and non-speech events, i.e., for transition a) from the end state of speech (*end_S*) to the start state of non-speech (*start_NS*), and b) from the end state of non-speech (*end_NS*) to the start state of speech (*start_S*).

To prepare training data containing transitions between speech and non-speech events, the data set from Sect. 3.4 was modified. At first, two recordings were chosen randomly from the artificial training set; one speech and one non-speech. After that, these two recordings were joined in a random order. The resulting recording then contained one of the two possible transitions (i.e., from speech to non-speech or from non-speech to speech) and was annotated automatically as follows:

1. The number of transition frames was derived from the input feature context window (25-1-25).
2. Only the 50 frames at the inner boundary of the two joined recordings were annotated as transitional, i.e., using 25 labels *stop_S* followed by 25 labels *start_NS* or 25 labels *stop_NS* followed by 25 labels *start_S*.
3. All other frames were marked as either speech or non-speech.

The results of the experiment with the context-based smoothing (see the fourth row of Table 1) show that this approach addresses the issue of an increase in $\delta_{2/3}$, which has emerged due to the use of the artificial training data (see the third row of Table 1). The value of $\delta_{2/3}$ was reduced from 0.34 s to 0.27 s. At the same time, a significant decrease in FAR, an increase in F-measure, and only a slight decrease in MR by 0.2% was achieved when compared to the previous experiment.

3.6. Evaluation on QUT-NOISE-TIMIT Corpus

The evaluation on QUT-NOISE-TIMIT corpus [23] shows the performance of the proposed approach in comparison with five approaches already presented in [23] and two techniques reaching the state-of-the-art results [24, 12]. The five approaches were: standardized VAD system ITU-T G.729 Annex B [25], standardized advanced front-end ETSI [26], Sohn's likelihood ratio test [27], Ramirez's long-term spectral divergence (LTSD) [28] and GMM based approach with use of MFCC features [23]. The latter two techniques were voice activity detection using subband noncircularity (SNC) [24] and complete-linkage clustering (CLC) for VAD [12].

The QUT-NOISE-TIMIT corpus was designed for training and evaluation of SAD systems in various noise scenarios and SNR levels. The data set combines clean speech from TIMIT corpus [29] with background noise recordings from QUT-NOISE data set [23]. The QUT-NOISE data set contains five types of background noises (scenarios: cafe, home, street, car, reverb) each from two different locations. Total amount of 600 hours were compiled and divided into two groups (A, B). Each group contains recordings from all scenarios in various SNR levels.

The training and testing protocols recommended for QUT-NOISE-TIMIT corpus presented in [23] were followed. The training was done with prior knowledge of target environment SNR; low noise (10, 15 dB), medium noise (0, 5 dB) and high noise (-10, -5 dB). However no prior knowledge of target environment scenario was utilized during the training phase. For each target SNR, group A was used for training and group B for testing and vice-versa. The proposed SAD module was trained as described in Sect. 3 with the exception of the use of artificial training data.

Figure 3 presents the comparison of the proposed SAD module and above introduced SAD approaches at three different noise levels: low, medium and high. In addition to MR and FAR, Half-Total Error Rate (HTER) was also evaluated. It is defined as equal-weighted average of MR and FAR. The obtained results show that our solution outperforms other SAD systems in low and medium noise conditions. The absolute reduction in HTER is over 2% over the previously best complete-linkage clustering approach. However, the HTER is approximately 2% worse in high noise conditions. The rest of the techniques are still being outperformed by a fair margin.

Our solution thus achieve state-of-the-art results in both low and medium noise conditions while staying competitive in high noise conditions on QUT-NOISE-TIMIT corpus.

4. RESULTS OF THE PROPOSED SAD APPROACH IN A REAL SPEECH TRANSCRIPTION SYSTEM

Given the findings and results from all previous experiments, the resulting SAD approach with the context-based smoothing was evaluated in a real speech-transcription system.

For this purpose, two test sets of Czech broadcasts were utilized. The first set represents 4 hours (22204 words) recorded from a Czech live news TV channel. Approximately 60% of its content consisted of speech segments. The length of the other set was 8 hours, it contained 7212 words, and speech frames formed only 10% of its content. This set represents broadcast of a Czech local radio station.

The transcription system employed an acoustic model based on a Hidden Markov Model - Deep Neural Network (HMM-DNN) hybrid architecture [30], where the baseline Gaussian Mixture Model (GMM) is trained as context-dependent, speaker-independent and contains 3886 physical states. The data for training of this model contained 270 hours of speech recordings. The parameters used for the DNN training were as follows: 5 hidden layers with a decreasing number of neurons per hidden layer (1024-1024-768-768-512),

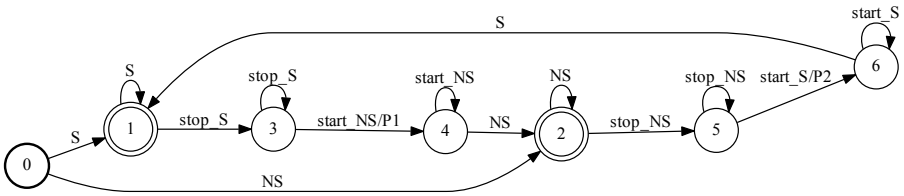


Fig. 2. The scheme of the WFST representing the context-based smoothing model.

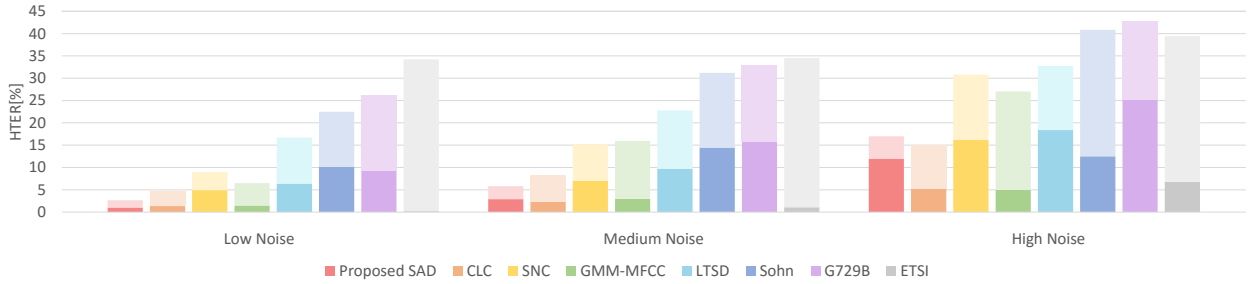


Fig. 3. Comparison of various SAD systems across QUT-NOISE-TIMIT corpus. HTER is defined as equal-weighted average of MR and FAR. The percentage contribution of MR and FAR to HTER bars is displayed by darker and lighter shades, respectively.

ReLU activation function, mini-batches of size 1024, 35 training epochs, learning rate 0.08. For signal parameterization, log filter banks were used with context windows of 5-1-5 and local normalization was employed within one-second windows.

The linguistic part of the system was composed of a lexicon and a language model. The lexicon contained 550k entries with multiple pronunciation variants and the language model was based on N-grams. For practical reasons (mainly with respect to the very large vocabulary size), the system used bigrams. However, 20 percent of all word-pairs actually include sequences containing three or more words, as the lexicon contains 4k multi-word collocations. The unseen bigrams are backed-off by Kneser-Ney smoothing.

4.1. Experimental Results

Within the performed experiments, both test sets were transcribed a) with and b) without the use of the SAD module. The obtained results are presented in Table 2, which contains values of Word Error Rate (WER) and Correctness (Corr) to show the transcription accuracy of the system. To measure computational demands with and without SAD, values of RTF (the ratio of the processing time to recording length) are also presented.

Table 2. Evaluation of the proposed SAD approach in a real speech transcription system.

Test set	live news TV channel		local radio station	
SAD module	Yes	No	Yes	No
WER [%]	12.4	12.7	14.0	17.9
Corr [%]	89.7	89.7	88.5	88.4
RTF	0.42	0.77	0.08	0.83

The obtained results indicate that the utilization of the proposed SAD approach was advantageous on both test sets. The yielded Corr and WER show that the SAD module limited the insertions coming from the non-speech parts and omitted hardly any speech parts.

The SAD module allowed the transcription system to operate with improved accuracy and, at the same time, RTF was almost two times, and more than ten times lower for the first and second test set, respectively. Of course, the reason for this difference is that the data in the second set contained fewer speech segments. Note that RTF of the SAD module itself is around 0.01 and all presented RTF values were measured using processor Intel Core i7-3770K @ 3.50GHz.

The transcription system complemented with SAD can also be utilized for online transcription without any major delay, because its latency is around 2 seconds.

5. CONCLUSIONS

In this paper, a new SAD approach suitable in offline as well as online speech transcription systems is proposed. The approach utilizes

- a DNN-based classifier;
- training data created artificially by mixing speech and non-speech recordings at various levels of SNR;
- a WFST-based decoder that smooths the output from DNN using a context-based model in which both the speech and non-speech events are represented as sequences of states.

The application of this approach to a real speech-recognition system leads to a) a slight decrease in WER, and b) significant reduction in RTF of the whole transcription process. The latter advantage is namely important for 24/7 monitoring of streams containing a large proportion of music (e.g., local radio stations), where the computational demands on the transcription system can be reduced dramatically.

6. ACKNOWLEDGEMENTS

This work was supported by the Technology Agency of the Czech Republic (Project No. TA04010199) and partly by the Student Grant Scheme 2017 of the Technical University in Liberec.

7. REFERENCES

[1] Georgios Evangelopoulos and Petros Maragos, "Speech event detection using multiband modulation energy.," in *INTERSPEECH*. 2005, pp. 685–688, ISCA.

[2] Bojan Kotnik, Zdravko Kacic, and Bogomir Horvat, "A multiconditional robust front-end feature extraction with a noise reduction procedure based on improved spectral subtraction algorithm.," in *INTERSPEECH*. 2001, pp. 197–200, ISCA.

[3] Houman Ghaemmaghami, Brendan Baker, Robert Vogt, and Sridha Sridharan, "Noise robust voice activity detection using features extracted from the time-domain autocorrelation function.," in *INTERSPEECH*. 2010, pp. 3118–3121, ISCA.

[4] Kaavya Sriskandaraja, Vidhyasaharan Sethu, Phu Ngoc Le, and Eliathamby Ambikairajah, "A model based voice activity detector for noisy environments," in *INTERSPEECH*, 2015, pp. 2297–2301.

[5] Neville Ryant, Mark Liberman, and Jiahong Yuan, "Speech activity detection on youtube using deep neural networks.," in *INTERSPEECH*. 2013, pp. 728–731, ISCA.

[6] Xiao-Lei Zhang and DeLiang Wang, "Boosted deep neural networks and multi-resolution cochleagram features for voice activity detection.," in *INTERSPEECH*. 2014, pp. 1534–1538, ISCA.

[7] Andreas Tsiartas, Theodora Chaspari, Nassos Katsamanis, Prasanta Kumar Ghosh, Ming Li, Maarten Van Segbroeck, Alexandros Potamianos, and Shrikanth Narayanan, "Multi-band long-term signal variability features for robust voice activity detection," in *INTERSPEECH*, 2013, pp. 718–722.

[8] Jeff Ma, "Improving the speech activity detection for the darpa rats phase-3 evaluation.," in *INTERSPEECH*. 2014, pp. 1558–1562, ISCA.

[9] Xiao-Lei Zhang and Ji Wu, "Deep belief networks based voice activity detection," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 4, pp. 697–710, April 2013.

[10] Jong Won Shin, Joon-Hyuk Chang, and Nam Soo Kim, "Voice activity detection based on statistical models and machine learning approaches," *Comput. Speech Lang.*, vol. 24, no. 3, pp. 515–530, July 2010.

[11] Tim Ng, Bing Zhang 0004, Long Nguyen, Spyros Matsoukas, Xinhui Zhou, Nima Mesgarani, Karel Vesely, and Pavel Matejka, "Developing a speech activity detection system for the darpa rats program.," in *INTERSPEECH*. 2012, pp. 1969–1972, ISCA.

[12] Houman Ghaemmaghami, David Dean, Shahram Kalantari, Sridha Sridharan, and Clinton Fookes, "Complete-linkage clustering for voice activity detection in audio and visual speech," in *INTERSPEECH*, 2015, pp. 2292–2296.

[13] George Saon, Samuel Thomas, Hagen Soltau, Sriram Ganapathy, and Brian Kingsbury, "The ibm speech activity detection system for the darpa rats program.," in *INTERSPEECH*. 2013, pp. 3497–3501, ISCA.

[14] Thad Hughes and Keir Mierle, "Recurrent neural networks for voice activity detection.," in *ICASSP*. 2013, pp. 7378–7382, IEEE.

[15] F. Eyben, F. Weninger, S. Squartini, and B. Schuller, "Real-life voice activity detection with lstm recurrent neural networks and an application to hollywood movies," in *ICASSP*, May 2013, pp. 483–487.

[16] Qing Wang, Jun Du, Xiao Bao, Zi-Rui Wang, Li-Rong Dai, and Chin-Hui Lee, "A universal vad based on jointly trained deep neural networks.," in *INTERSPEECH*. 2015, pp. 2282–2286, ISCA.

[17] Samuel Thomas, George Saon, Maarten Van Segbroeck, and Shrikanth S. Narayanan, "Improvements to the ibm speech activity detection system for the darpa rats program.," in *ICASSP*. 2015, pp. 4500–4504, IEEE.

[18] Chao Gao, Guruprasad Saikumar, Saurabh Khanwalkar, Avi Herscovici, Anoop Kumar, Amit Srivastava, and Premkumar Natarajan, "Online speech activity detection in broadcast news.," in *INTERSPEECH*. 2011, pp. 2637–2640, ISCA.

[19] Hoon Chung, Sung Joo Lee, and Yunkeun Lee, "Endpoint detection using weighted finite state transducer.," in *INTERSPEECH*. 2013, pp. 700–703, ISCA.

[20] Baiyang Liu, Björn Hoffmeister, and Ariya Rastrow, "Accurate endpointing with expected pause duration," in *INTERSPEECH*. 2015, pp. 2912–2916, ISCA.

[21] M. H. Moattar and M. M. Homayounpour, "A simple but efficient real-time voice activity detection algorithm," in *Signal Processing Conference, 2009 17th European*, Aug 2009, pp. 2549–2553.

[22] Okko Johannes Räsänen, Unto Kalervo Laine, and Toomas Altosaar, "An improved speech segmentation quality measure: the r-value.," in *INTERSPEECH*, 2009, pp. 1851–1854.

[23] David Dean, Sridha Sridharan, Robert Vogt, and Michael Mason, "The qut-noise-timit corpus for the evaluation of voice activity detection algorithms.," in *INTERSPEECH*. 2010, pp. 3110–3113, ISCA.

[24] S. Wisdom, G. Okopal, L. Atlas, and J. Pitton, "Voice activity detection using subband noncircularity," in *ICASSP*, April 2015, pp. 4505–4509.

[25] A. Benyassine, E. Shlomot, H. Y. Su, D. Massaloux, C. Lamblin, and J. P. Petit, "Itu-t recommendation g.729 annex b: a silence compression scheme for use with g.729 optimized for v.70 digital simultaneous voice and data applications," *IEEE Communications Magazine*, vol. 35, no. 9, pp. 64–73, Sep 1997.

[26] Jin-Yu Li, Bo Liu, Ren-Hua Wang, and Li-Rong Dai, "A complexity reduction of etsi advanced front-end for dsr," in *ICASSP*, May 2004, vol. 1, pp. I–61–4 vol.1.

[27] Jongseo Sohn, Nam Soo Kim, and Wonyong Sung, "A statistical model-based voice activity detection," *IEEE Signal Processing Letters*, vol. 6, no. 1, pp. 1–3, Jan 1999.

[28] Javier Ramrez, Jose C. Segura, Carmen Bentez, Angel De La Torre, and Antonio Rubio, "Efficient voice activity detection algorithms using long-term speech information," *Speech Communication*, vol. 42, pp. 3–4, 2004.

[29] William M. Fisher, George R. Doddington, and Kathleen M. Goudie-Marshall, "The DARPA Speech Recognition Research Database: Specifications and Status," in *Proceedings of DARPA Workshop on Speech Recognition*, 1986, pp. 93–99.

[30] G.E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 30 –42, jan. 2012.

Using X-vectors for Speech Activity Detection in Broadcast Streams

Lukas Mateju, Frantisek Kynych, Petr Cerva, Jindrich Zdansky, Jiri Malek

Faculty of Mechatronics, Informatics and Interdisciplinary Studies,
Technical University of Liberec, Studentska 2, 461 17 Liberec, Czech Republic
{lukas.mateju, frantisek.kynych, petr.cerva, jindrich.zdansky, jiri.malek}@tul.cz

Abstract

A new approach to speech activity detection (SAD) is presented in this work. It allows us to reduce the complexity and computation demands, namely in services that process streaming speech, where a SAD module usually forms the first block of the data pipeline (e.g., in a platform for 24/7 broadcast transcription). Our approach utilizes x-vectors as input features so that, within the subsequent pipeline stages, these embedding instances can also directly be employed for speaker diarization and recognition. The x-vectors are extracted by feed-forward sequential memory network (FSMN), allowing for modeling long-time dependencies; they thus form an input into a computationally undemanding binary classifier, whose output is smoothed by a decoder. Evaluation is performed on the standardized QUT-NOISE-TIMIT dataset as well as on broadcast data with large portions of music and background noise. The former data allows for comparison with other existing approaches. The latter shows the performance in terms of word error rate (WER) and reduction in real-time factor (RTF) of the transcription process.
Index Terms: x-vectors, speech activity detection, streamed data processing, deep neural networks

1. Introduction

X-vectors allow mapping of variable-length utterances to fixed-dimensional embeddings. They can be extracted using various deep neural network (DNN) architectures and provide robust representations when a large amount of training data is used. It has recently been shown that these embeddings are able to encode various attributes of an input utterance including its length, channel information, speaker's gender, speaking rate or even spoken content [1]. In [2], x-vectors were also successfully applied to the spoken language recognition task.

However, x-vectors were originally crafted for speaker recognition [3]. The speaker characteristics encoded in the embedding can also be utilized for speaker diarization (SD). In this task, x-vectors are clustered using various techniques [4, 5]. The SD systems usually operate just over speech segments of the input data. In some SD evaluation tracks [6], these segments are extracted manually; in other tracks, as well as in practice, an automatic SAD module must be used.

To the best of our knowledge, the existing SAD approaches do not utilize x-vectors (see also Sec. 2). Therefore, two systems must be employed over different features (one for SAD and one for SD) in real SD applications. This approach increases computation demands and the complexity of the whole speech processing pipeline, which is undesirable, particularly in systems designed for processing streamed data. In our case, an example of such a system is a platform for 24/7 monitoring of various TV and radio streams, namely in Slavic languages (see our multilingual radio monitoring application[1]). In this type of application, the SAD module forms the first block of the data-processing pipeline and its output, consisting of the speech segments, is used for a) speaker diarization (followed by speaker recognition) and b) speech transcription. In the latter task, the SAD module allows us to reduce the computation demands dramatically as some broadcast streams contain large amounts of non-speech parts, such as songs or advertisements.

In this work, we extend our previous investigations to the use of x-vectors for speech activity (and overlapped speech) detection [7] by presenting a complete SAD scheme, which is particularly suitable for the above-mentioned frame-wise processing. The use of x-vectors emphasizes one of the advantages of our approach because they can also be directly employed for speaker diarization and recognition in the subsequent data pipeline stages. Experimental results presented in Sec. 5 show that our approach yields results that are similar to (or even better than) the existing state-of-the-art SAD methods.

2. Related work

Most commonly, speech activity detection is run in two consecutive phases: feature extraction followed by speech/non-speech classification. In the former phase, the more classical approaches utilize, e.g., zero-crossing rate [8], energy [9], or auto-correlation function [10]. Over the years, more complex features including multi-resolution cochleagram features [11], Mel-frequency cepstral coefficients (MFCCs) [12], or pitch related features [13] have been applied with great success. Furthermore, bottleneck features extracted from DNNs have been proposed in [14]. In practice, various combinations of individual features are often used to achieve the best possible results.

In the latter phase, different classifiers can be employed, including support vector machines [15] or Gaussian mixture models (GMMs) [16, 17]. In recent years, various DNN architectures, such as fully connected (FC) feed-forward DNNs [12] or convolutional neural networks (CNNs) [18] have frequently been applied. The modeling power of recurrent neural networks (RNNs) has been exploited as well [19]. Specifically, the long short-term memory (LSTM) RNNs [20, 21] have recently gained a lot of popularity. More complex approaches, e.g., boosted DNNs [11] or convolutional LSTM neural networks [22], have also been presented. Moreover, convolutional gated recurrent unit (GRU) RNNs [23] have been utilized successfully. Recently, an adaptive context attention model was proposed in [24]. Finally, unsupervised approaches, such as rVAD [25], have been applied as well.

To improve the accuracy of the speech activity detection, the outputs from a given classifier can also be smoothed. Different techniques, such as the Viterbi decoder [12], weighted finite-state transducers (WFSTs) [26], or temporal smoothing layers, CNN or RNN (with bidirectional GRU) ones [27], have been suggested for this purpose.

[1] https://tul-speechlab.gitlab.io/

Table 1: *The structure of FSMN-based x-vector extractor.*

Layer	Layer context	Total context	Input x output
FSMN 1	$\ell \pm 80$	161	40×1024
FSMN 2	$\ell \pm 4$	169	1024×768
FSMN 3	$\ell \pm 4$	177	768×512
FSMN 4	$\ell \pm 4$	185	512×384
FSMN 5	$\ell \pm 4$	193	384×256
FSMN 6	$\ell \pm 4$	201	256×128
FC 1	ℓ	201	128×128
Pooling	$\ell \pm 20$	241	$(41 \cdot 128) \times 128$
FC 2	ℓ	241	128×128
Softmax	–	241	$128 \times Nspeakers$

3. Proposed approach

The proposed approach consists of three consecutive steps, which are described in detail in the following subsections:

1. Extraction of x-vectors using DNN.
2. Classification of x-vectors into two classes.
3. Smoothing the output from the classifier by a decoder.

3.1. X-vectors extraction

In the first step, a vectorized variant of the FSMN [28] is employed for x-vectors extraction. This architecture allows us to model long-time dependencies in the input signal similar to RNNs, but it eliminates the recursion by adding several memory blocks with trainable weight coefficients into each layer of a standard feed-forward FC DNN. The memory blocks use a tapped-delay line structure to encode the long context information into a fixed-size representation. That means that the FSMN layers are built on top of the context of the earlier layers; the final context is thus a sum of the partial ones.

The structure of FSMN used in this work is described in Table 1, where the symbol ℓ denotes the current frame, on which the temporal context is centered. The pooling layer computes only the means of the frames (omitting the variances) in the context of 41 consecutive frames. In all neurons, exponential linear unit (ELU) is used as the activation function. On the input, each frame of the signal is represented by 40 log filter bank coefficients (FBCs) computed from 25-ms-long frames with frame-shifts of 12.5 ms each. Table 1 shows that the extractor operates with a total context of 241 frames, which corresponds to 3.0125 seconds. Note that, within the context of the first layer (i.e., 161 frames), cepstral mean subtraction (CMS) is applied and the x-vectors are extracted after the pooling layer.

3.2. Binary classification of x-vectors

In the second step, the extracted embeddings are utilized by a binary DNN-based classifier that produces probabilities for the speech/non-speech classes. The basic and computationally undemanding architecture of the classifier employs two feed-forward FC hidden layers with 128 and 64 neurons. In Sec. 5.2, we also present results for other topologies.

3.3. Smoothing using WFST-based decoder

For smoothing the output from the classifier, a WFST-based decoder is employed in the last step. Its advantage is that it represents a general smoothing concept and allows us to model various sorts of smoothing approaches by merely choosing the corresponding transduction model and the respective semi-ring.

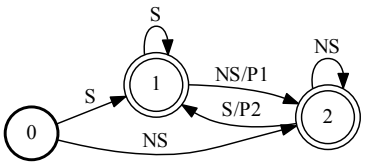


Figure 1: *The transducer (acceptor) representing the basic transduction model used for speech/non-speech detection.*

The transduction model employed in this work is depicted in Figure 1 and corresponds to the basic scheme that we have proposed for SAD over FBCs with feed-forward FC DNN and WFST-based decoder [29]. It consists of three states. The first state, denoted by 0, is the initial state. The transitions between states 1 and 2 emit the speech/non-speech labels and are penalized by penalty factors P1 and P2, respectively. Their values (100 and 100) were determined in several experiments not presented in this paper. More details about the decoding process with WFST can be found in [30], where we have also applied this type of smoothing to on-line language identification.

4. Experimental setup

4.1. Development data

For development purposes, a dataset consisting of 6 hours of TV and radio recordings in several Slavic languages (Czech, Slovak, Polish, and Russian) was constructed. These recordings contain clean speech segments and segments with music, background noise, jingles, and advertisements. Annotations of this data were at first created automatically (by our previous system) and later corrected and fine-tuned by hand. In total, 70% of all frames were marked as containing speech.

4.2. Evaluation metrics

Within this work, frame error rate (FER), miss rate (MR), and false alarm rate (FAR) are utilized to evaluate the overall frame accuracy of SAD [12]. Additionally, the quality of change-point detection (between speech and non-speech segments) given the alignment between detected and reference boundaries is expressed by the F-measure (Fm).

5. Experimental evaluation

5.1. X-vector extraction

The opening set of experiments is focused on the first step of the proposed approach: three different x-vector extractors based on the FSMN topology are investigated. In the first case, speech x-vectors (denoted by xv_s below) were calculated for all (7237) individual training speakers as usual. The training data consisted of Voxceleb2 [31], "train-360-clean" part of LibriSpeech [32] and 121 hours of clean Czech microphone recordings belonging to 922 speakers. In the second case of the xv_{s+n} extractor, the set of target speakers was extended by the noise class defined over the train part of the CHiME-4 dataset [33]. Finally, one special class representing music was also added. The resulting extractor is denoted by xv_{s+n+m}; the music data used for training contain 31 hours of various music recordings.

Given all these three types of extractors, three two-layer binary classifiers (see also Sec. 5.2) have been trained. The train-

Table 2: *Results of different x-vector extractors in comparison with FBCs on the development dataset.*

approach	FER[%]	MR[%]	FAR[%]	Fm[%]
xv_s	2.6	0.8	7.1	52.9
xv_{s+n}	**2.2**	0.7	6.0	**59.3**
xv_{s+n+m}	**2.2**	0.8	**5.8**	58.8
FBCs + DNN	2.5	**0.5**	7.7	54.3
FBCs + FSMN	2.6	**0.5**	8.2	53.2

ing dataset consists of 30 hours of clean speech, 30 hours of music and 30 hours of artificially mixed speech and music/noise recordings according to randomly chosen signal-to-noise ratio (SNR). All of these recordings are also concatenated in a random order to contain speech/non-speech transitions. For annotations, music recordings and the segments with SNR lower than 0 dB are labeled as non-speech and the rest as speech. Finally, the outputs of the classifiers are smoothed by the WFST-based decoder (see Sec. 3.3 for more details).

The results are summarized in Table 2. They show that all three x-vector extractors lead to a very low level of MR as well as FER. In terms of F-measure, the xv_{s+n} extractor yields the best results as the additional noise class improves the performance for more noisy segments. On the other hand, additional music class does not yield any further improvements.

The last two rows in Table 2 are dedicated to comparison with the approach we presented in [29]. It analogically employed a DNN-based binary classifier and a WFST-based decoder smoothing the outputs of the DNN. The main difference is that the classifier was trained directly on FBCs. The DNN has five hidden layers, each with 128 neurons, and the ReLU activation function is utilized. To allow fair comparison with the SAD module proposed in this work, the same context is used, i.e., the input feature vector is formed by concatenating 50 previous frames, the current frame (39-dimensional FBCs), and the 50 following frames. Local normalization has also been performed as for the x-vectors within a two-second window.

Finally, we also trained an FSMN-based classifier with the corresponding parameters.

The results in the last two rows in Table 2 show that the DNN- and FSMN-based baselines perform comparably but yield outcomes that are overall worse than those of the x-vector systems (the only exception is given by the lower MR values).

5.2. Binary classification

To evaluate the second step in the proposed approach, several different NN architectures varying in the complexity have been explored for binary classification.

The basic topology with 2 hidden layers (denoted as DNN-2HL) is described in Sec. 3.2 and was used in all previous experiments. The less complex variants include a) simple NN without any hidden layer (NN-0HL) and b) NN with one hidden layer with 128 neurons (NN-1HL). The latter network proved to be efficient for detecting overlapped speech in our previous study [7]. On the contrary, the more complex topology is represented by DNN with five hidden layers (DNN-5HL), each with 128 neurons (it corresponds to the classifier used in the previous experiments over FBCs). This DNN was trained with a) zero input context and b) 0.5-second context window (i.e., 25 previous frames, the current frame, and 25 following frames). In addition, two more complex architectures, time-delay neural network (TDNN) and FSMN, both having five hidden layers with 128 neurons per layer, have also been evaluated.

Table 3: *The performance of various binary classifiers over xv_{s+n} extractor.*

classifier	FER[%]	MR[%]	FAR[%]	Fm[%]
zero input context				
NN-0HL	**2.2**	0.9	**5.5**	59.7
NN-1HL	**2.2**	0.8	5.9	59.8
DNN-2HL	**2.2**	0.7	6.0	59.3
DNN-5HL	2.4	0.6	7.0	54.1
0.5-second input context				
DNN-2HL	2.4	0.7	6.8	52.6
TDNN-5HL	2.8	**0.5**	8.4	57.6
FSMN-5HL	2.4	0.6	7.0	**60.8**

Table 4: *The effect of smoothing the binary classifier's output on the performance of SAD.*

smoothing	FER[%]	MR[%]	FAR[%]	Fm[%]
none	4.2	2.5	8.6	0.5
MA 1 s	2.8	0.9	7.8	32.2
MA 2 s	2.8	0.6	8.4	48.5
MA 3 s	3.0	**0.5**	9.3	40.0
WFST	**2.2**	0.8	**5.9**	**59.8**

Based on the obtained results (see Table 3), several conclusions can be made. First, no additional input context is needed for classification as the x-vectors already encode long context information in the FSMN topology. Second, the deeper DNNs do not yield any further significant improvements over the corresponding shallower architectures. Third, the FSMN-based classifier achieves the highest F-measure at the cost of worse FER and much higher computational demands. To sum it up, we have chosen the NN-1HL classifier for further experimental evaluation as it represents a compromise between the SAD performance and computational demands.

5.3. The effect of smoothing

The third experiment is aimed at the last step of the proposed approach, i.e., smoothing the output of the classifier. Here, we show the importance of the WFST-based smoothing by comparing it to smoothing with moving average (MA; with different lengths of the window) as well as to no smoothing at all.

The results are presented in Table 4. They clearly show that the WFST-based decoder yields the best results in FER, FAR, and F-measure by a large margin. The necessity of smoothing is demonstrated in the first row, corresponding to no smoothing. In this case, the non-stop changes between speech and non-speech segments resulted in an extremely small value of the F-measure.

6. ASR results

Given the results from evaluation on the development data, the resulting SAD approach has also been evaluated in a real speech-transcription system. This system utilizes an FSMN-based acoustic model, an n-gram language model and a lexicon containing 400k of the most frequent Czech words. Two distinct Czech datasets have been used. The first one consists of

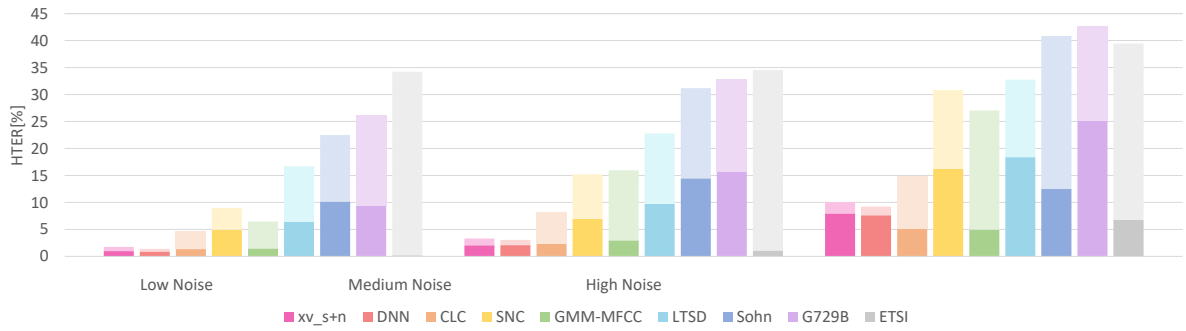


Figure 2: *A comparison among different SAD approaches across the QUT-NOISE-TIMIT corpus. The contributions of MR and FAR to HTER bars are displayed by darker and lighter shades, respectively.*

Table 5: *The influence of the proposed SAD approach on the performance of an ASR system.*

SAD	WER	RTF	WER	RTF
	TV news		local radio	
none	14.7	0.67	65.2	0.76
xv_{s+n}	**11.0**	0.33	**13.6**	**0.06**
FBCs + DNN	11.1	0.33	14.3	**0.06**
xv_{s+n} with red. context	11.4	**0.32**	14.4	**0.06**

four hours (22k words) of recordings from a live news TV channel, and approximately 60% of its content is speech. The other dataset comes from a local radio station. Its length is 8 hours (7k words), and only 10% of its duration is formed by speech.

The achieved results in terms of WER and RTF are presented in the first three rows in Table 5. They show that the proposed SAD module has slightly outperformed the similar module based on DNN and FBCs. The decrease in WER indicates how effectively the insertions are limited in non-speech parts and hardly omit any speech. Furthermore, the RTF has improved 2 and 12 times on the news and radio test sets, respectively. The RTF value of the SAD approach itself is 0.01. Note that the presented RTF values have been measured using processor Intel Core i7-3770K @ 3.50GHz.

Finally, it should also be noted that the latency of the SAD module is 3.2 seconds (out of which 1.7 seconds is caused by the decoder). This value may be considered too large for online applications (e.g., subtitling). For this reason, we have tried to reduce the latency by 0.75 seconds, taking a quarter of the context for the first fixed layer of the x-vector extractor. This operation has led to just a slight and acceptable decrease in WER (see the last row in Table 5).

7. Results for QUT-NOISE-TIMIT

A QUT-NOISE-TIMIT [17] corpus has been utilized to compare the proposed approach with five systems already presented in [17], two newer techniques [34, 35], and the DNN-based classifier with WFST-based smoothing described in Sec. 5.1. The five original approaches were: standardized advanced front-end ETSI [36], standardized VAD system ITU-T G.729 Annex B [37], Sohn's likelihood ratio test [38], Ramirez's long-term spectral divergence (LTSD) [39] and GMM-based approach over MFCCs [17]. The latter two techniques were VAD using subband noncircularity (SNC) [34] and complete-linkage clustering (CLC) for VAD [35].

The training and testing protocols for QUT-NOISE-TIMIT corpus were followed, as recommended in [17]. During the training, the only prior knowledge given to the system was the target environment SNR, low noise (10, 15 dB), medium noise (0, 5 dB) or high noise (-10, -5 dB). The proposed SAD approach was trained as described in Sec. 3, i.e., xv_{s+n} extractor followed by a classifier with just one hidden layer and with WFST-based smoothing applied. The one exception was the use of only QUT-NOISE-TIMIT data for training of the classifier.

Figure 2 depicts the obtained results in all target SNR environments: low, medium and high. In addition to MR and FAR, half-total error rate (HTER) has also been reported. It is defined as equal-weighted average of MR and FAR. The results show, that in all noise conditions, the proposed approach outperforms most other SAD systems by a large margin. The only exception is the DNN-based approach over FBCs, which yields a slightly better performance.

8. Conclusions

In this paper, a new SAD approach suitable for processing streamed data is proposed. The method utilizes FSMN-based x-vectors as the input features to a computationally undemanding binary classifier (with only a single hidden layer), whose output is smoothed by a WFST-based decoder.

The results achieved on the development set and the QUT-NOISE-TIMIT corpus show that the proposed method yields state-of-the-art results and is capable of outperforming many other approaches while operating in a frame-wise mode and with possibility of on-line use. It also has a similar performance as a directly comparable approach based on DNN-based classifier (but over FBCs) with WFST-based smoothing. However, the main advantage of our method in comparison with the DNN-based approach is implied by the fact that the x-vectors used for SAD can also be simply employed for SD or recognition in the subsequent stages of the data-processing pipeline.

Finally, additional experiments performed in an ASR system have proved that the use of the method allows us to reduce the WER value and significantly improve the RTF level of the transcription process.

9. Acknowledgements

This work was supported by the Technology Agency of the Czech Republic (project No. TH03010018), and by the Student Grant Competition of the Technical University of Liberec under project No. SGS-2019-3017.

10. References

[1] D. Raj, D. Snyder, D. Povey, and S. Khudanpur, "Probing the information encoded in x-vectors," in *ASRU 2019, Singapore*, 2019, pp. 726–733.

[2] D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, D. Povey, and S. Khudanpur, "Spoken language recognition using x-vectors," in *Odyssey 2018, Les Sables d'Olonne, France*, 2018, pp. 105–111.

[3] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *ICASSP 2018, Calgary, Canada*, 2018, pp. 5329–5333.

[4] D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, and A. McCree, "Speaker diarization using deep neural network embeddings," in *ICASSP 2017, New Orleans, USA*, 2017, pp. 4930–4934.

[5] M. Diez, L. Burget, F. Landini, and J. Cernocky, "Analysis of speaker diarization based on bayesian HMM with eigenvoice priors," *IEEE/ACM Transactions Audio, Speech & Language Processing*, vol. 28, pp. 355–368, 2020.

[6] N. Ryant, K. Church, C. Cieri, A. Cristia, J. Du, S. Ganapathy, and M. Liberman, "The second DIHARD diarization challenge: Dataset, task, and baselines," in *Interspeech 2019, Graz, Austria*, 2019, pp. 978–982.

[7] J. Malek and J. Zdansky, "Voice-activity and overlapped speech detection using x-vectors," in *TSD 2020, Brno, Czech Republic*, 2020, pp. 366–376.

[8] B. Kotnik, Z. Kacic, and B. Horvat, "A multiconditional robust front-end feature extraction with a noise reduction procedure based on improved spectral subtraction algorithm," in *Interspeech 2001, Aalborg, Denmark*, 2001, pp. 197–200.

[9] G. Evangelopoulos and P. Maragos, "Speech event detection using multiband modulation energy," in *Interspeech 2005, Lisbon, Portugal*, 2005, pp. 685–688.

[10] H. Ghaemmaghami, B. Baker, R. Vogt, and S. Sridharan, "Noise robust voice activity detection using features extracted from the time-domain autocorrelation function," in *Interspeech 2010, Makuhari, Japan*, 2010, pp. 3118–3121.

[11] X. Zhang and D. Wang, "Boosted deep neural networks and multi-resolution cochleagram features for voice activity detection," in *Interspeech 2014, Singapore*, 2014, pp. 1534–1538.

[12] N. Ryant, M. Liberman, and J. Yuan, "Speech activity detection on youtube using deep neural networks," in *Interspeech 2013, Lyon, France*, 2013, pp. 728–731.

[13] Y. Shao and Q. Lin, "Use of pitch continuity for robust speech activity detection," in *ICASSP 2019, Calgary, Canada*, 2018, pp. 5534–5538.

[14] L. Ferrer, M. Graciarena, and V. Mitra, "A phonetically aware system for speech activity detection," in *ICASSP 2016, Shanghai, China*, 2016, pp. 5710–5714.

[15] J. W. Shin, J. Chang, and N. S. Kim, "Voice activity detection based on statistical models and machine learning approaches," *Computer Speech & Language*, vol. 24, pp. 515–530, 2010.

[16] T. Ng, B. Zhang, L. Nguyen, S. Matsoukas, X. Zhou, N. Mesgarani, K. Vesely, and P. Matejka, "Developing a speech activity detection system for the DARPA RATS program," in *Interspeech 2012, Portland, USA*, 2012, pp. 1969–1972.

[17] D. Dean, S. Sridharan, R. Vogt, and M. Mason, "The QUT-NOISE-TIMIT corpus for the evaluation of voice activity detection algorithms," in *Interspeech 2010, Makuhari, Japan*, 2010, pp. 3110–3113.

[18] G. Saon, S. Thomas, H. Soltau, S. Ganapathy, and B. Kingsbury, "The IBM speech activity detection system for the DARPA RATS program," in *Interspeech 2013, Lyon, France*, 2013, pp. 3497–3501.

[19] T. Hughes and K. Mierle, "Recurrent neural networks for voice activity detection," in *ICASSP 2013, Vancouver, Canada*, 2013, pp. 7378–7382.

[20] F. Eyben, F. Weninger, S. Squartini, and B. W. Schuller, "Real-life voice activity detection with LSTM recurrent neural networks and an application to hollywood movies," in *ICASSP 2013, Vancouver, Canada*, 2013, pp. 483–487.

[21] Q. Lin, T. Li, and M. Li, "The DKU speech activity detection and speaker identification systems for fearless steps challenge phase-02," in *Interspeech 2020, Shanghai, China*, 2020, pp. 2607–2611.

[22] R. Zazo, T. N. Sainath, G. Simko, and C. Parada, "Feature learning with raw-waveform CLDNNs for voice activity detection," in *Interspeech 2016, San Francisco, USA*, 2016, pp. 3668–3672.

[23] A. Vafeiadis, E. Fanioudakis, I. Potamitis, K. Votis, D. Giakoumis, D. Tzovaras, L. Chen, and R. Hamzaoui, "Two-dimensional convolutional recurrent neural networks for speech activity detection," in *Interspeech 2019, Graz, Austria*, 2019, pp. 2045–2049.

[24] J. Kim and M. Hahn, "Voice activity detection using an adaptive context attention model," *IEEE Signal Processing Letters*, vol. 25, pp. 1181–1185, 2018.

[25] Z. Tan, A. K. Sarkar, and N. Dehak, "rvad: An unsupervised segment-based robust voice activity detection method," *Computer Speech & Language*, vol. 59, pp. 1–21, 2020.

[26] H. Chung, S. J. Lee, and Y. Lee, "Endpoint detection using weighted finite state transducer," in *Interspeech 2013, Lyon, France*, 2013, pp. 700–703.

[27] J. Heitkaemper, J. Schmalenstroeer, and R. Haeb-Umbach, "Statistical and neural network based speech activity detection in non-stationary acoustic environments," in *Interspeech 2020, Shanghai, China*, 2020, pp. 2597–2601.

[28] S. Zhang, C. Liu, H. Jiang, S. Wei, L. Dai, and Y. Hu, "Feedforward sequential memory networks: A new structure to learn long-term dependency," *CoRR*, vol. abs/1512.08301, 2015.

[29] L. Mateju, P. Cerva, J. Zdansky, and J. Malek, "Speech activity detection in online broadcast transcription using deep neural networks and weighted finite state transducers," in *ICASSP 2017, New Orleans, USA*, 2017, pp. 5460–5464.

[30] P. Cerva, L. Mateju, J. Zdansky, R. Safarik, and J. Nouza, "Identification of related languages from spoken data: Moving from off-line to on-line scenario," *Computer Speech & Language*, vol. 68, 2021.

[31] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Interspeech 2018, Hyderabad, India*, 2018, pp. 1086–1090.

[32] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *ICASSP 2015, South Brisbane, Australia*, 2015, pp. 5206–5210.

[33] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech & Language*, vol. 46, pp. 535–557, 2017.

[34] J. Ramirez, J. C. Segura, M. C. Benitez, A. de la Torre, and A. J. Rubio, "Efficient voice activity detection algorithms using long-term speech information," *Speech Communication*, vol. 42, pp. 271–287, 2004.

[35] H. Ghaemmaghami, D. Dean, S. Kalantari, S. Sridharan, and C. Fookes, "Complete-linkage clustering for voice activity detection in audio and visual speech," in *Interspeech 2015, Dresden, Germany*, 2015, pp. 2292–2296.

[36] J. Li, B. Liu, R. Wang, and L. Dai, "A complexity reduction of ETSI advanced front-end for DSR," in *ICASSP 2004, Montreal, Canada*, 2004, pp. 61–64.

[37] A. Benyassine, E. Shlomot, H. Y. Su, D. Massaloux, C. Lamblin, and J. P. Petit, "ITU-T recommendation G.729 Annex B: a silence compression scheme for use with G.729 optimized for V.70 digital simultaneous voice and data applications," *IEEE Communications Magazine*, vol. 35, pp. 64–73, 1997.

[38] J. Sohn and W. Sung, "A voice activity detector employing soft decision based noise spectrum adaptation," in *ICASSP 1998, Seattle, USA*, 1998, pp. 365–368.

[39] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *IEEE Signal Processing Letters*, vol. 6, 1999.

ROBUST AUTOMATIC RECOGNITION OF SPEECH WITH BACKGROUND MUSIC

Jiri Malek, Jindrich Zdansky and Petr Cerva

Faculty of Mechatronics, Informatics, and Interdisciplinary Studies, Technical University of Liberec,
Studentská 2, 461 17 Liberec, Czech Republic.

jiri.malek@tul.cz

ABSTRACT

This paper addresses the task of Automatic Speech Recognition (ASR) with music in the background, where the accuracy of recognition may deteriorate significantly. To improve the robustness of ASR in this task, e.g. for broadcast news transcription or subtitles creation, we adopt two approaches: 1) multi-condition training of the acoustic models and 2) denoising autoencoders followed by acoustic model training on the preprocessed data. In the latter case, two types of autoencoders are considered: the fully connected and the convolutional network.

Presented experimental results show that all the investigated techniques are able to improve the recognition of speech distorted by music significantly. For example, in the case of artificial mixtures of speech and electronic music (low Signal-to-Noise Ratio (SNR) of 0 dB), we achieved absolute improvement of accuracy by 35.8%. For real-world broadcast news and a high SNR (about 10 dB), we achieved improvement by 2.4%. The important advantage of the studied approaches is that they do not deteriorate the accuracy in scenarios with clean speech (the decrease is about 1%).

***Index Terms*—** Robust recognition, background music, feature enhancement, denoising autoencoder, multi-condition training.

1. INTRODUCTION

Nowadays, the research in automatic speech recognition (ASR) is focused on robustness of the performance with respect to difficult environmental conditions. These include, e.g., distant microphones, concurrent speech or background interference. In some applications, such as online $24/7$ monitoring of broadcast media, one of the most often encountered background interferences is music.

Two basic approaches exist which introduce the robustness to background interference into ASR. The first approach consists in utilization of the *multi-condition training* of the acoustic models. Here, the distorted speech signals are included in the training set, i.e., the model incorporates knowledge on the possible interference. The disadvantage here is the difficulty of including all possible noise types in the training data, which are later encountered within test environments [1]. Considering environmental noise, this approach was reported to obtain high performance in [2]. Besides additive noise, this technique was demonstrated to be beneficial for reverberated speech in [3, 4].

The other approach is to perform input speech (or feature) preprocessing, in order to separate the speech from the interference. The ASR is performed on the enhanced signal / features. An efficient speech separation can be achieved using *denoising autoencoders*, such as those proposed for environmental noises in [5]. The benefits of autoencoders for ASR was shown in [6]. Here, the car and factory noises were considered. The performance of multichannel autoencoders was demonstrated on the Chime-2 challenge [7] datasets in [8].

The front-end preprocessing usually introduces distortions into enhanced data, which are not observed by the acoustic model trained on the clean data. To mitigate, the enhancement is usually applied on both training and test data and a new acoustic model is trained on the enhanced dataset. This is shown for environmental noises, e.g., in [9].

When comparing the two above-mentioned approaches, some studies get superior results using front-end denoising [10], while others favor the multi-condition training [2].

Focusing specifically on separation of background music, Nonnegative Matrix Factorization (NMF, [11, 12, 13]) is often utilized. A direct application to robust ASR was proposed in [14], introducing a probabilistic approach based on a catalog of prepared music samples. The utilization of autoencoders for music-robust ASR was proposed in [15]. That paper compares utilization of the fully connected and convolutional networks. It demonstrates that the autoencoder is capable of learning features to discriminate between music and speech. Moreover, the method is shown to be largely language-independent.

Relation to prior work: The above mentioned techniques are usually employed in the context of environmental background noise. In this paper, we specifically focus on background music. We extend the analysis of the denoising autoencoders from [15] and compare the autoencoders directly to multi-condition training [2]. We aim to determine more suitable approach for music-robust ASR. Compared to [15], where autoencoders were trained for a specific musical piece, we train more general models using broad range of artificial mixtures of speech and various music. Considered genres range from classical music to electronic tunes. We study the robustness of the models with respect to unseen test conditions (varied music genres and energy of background music) and confirm the functionality on real-world radio broadcast shows.

2. PROBLEM FORMULATION AND DATA DESCRIPTION

We focus on robustness of ASR to music present in the background of the speech. All of the considered training data are generated artificially, by summation of the speech and music signal. We analyze different scenarios, with respect to average Signal-to-Noise Ratio (SNR) and the included music genres.

We consider a Large Vocabulary Continuous Speech Recognition (LVCSR) task. Due to the data most readily available to us, we focus on Czech language, without any loss of generality to the investigated problems. Our training set consists of 132 hours of Czech speech.

This work was supported by the Technology Agency of the Czech Republic (Project No. TA04010199).

Table 1. Setup of the training set for multi-style acoustic models and respective autoencoders

Dataset (genre)	N	SNR levels	Music styles included
Piano 1	3	clean, $10, 5, 0$	Classical piano
Piano 2	7	clean, $10, 5, 0, -5, -10, -15, -20$	Classical piano
Electronic	3	clean, $10, 5, 0$	Ambient, dance, down-tempo, chillout or idm

Table 2. Setup of the artificially generated test sets

Dataset (genre)	SNR levels	Music styles included
Clean	clean	None
Test:Piano	$10, 0, -10, -20$	Classical piano
Test:Violin	$10, 0, -10, -20$	Piano and violin compositions
Test:Electro	$10, 5, 0, -5$	Ambient, dance, down-tempo, chillout or idm

The music we utilize in generation of the training dataset originates in a database of free music tracks at the Free Music Archive [16]. We use the *Piano* tracks (duration 33 minutes) and a broad set of *Electronic* music (667 minutes). The latter set consists of genres such as ambient, dance, down-tempo, chillout or idm. The piano music provides the easier scenario; the music covers partly different frequency bands than the speech, with only a single instrument present. The mixtures are intelligible even for very low SNR. As a more difficult scenario, we select the electronic music, because it resembles the background music of the TV shows.

3. PROPOSED ROBUSTNESS-INTRODUCING TECHNIQUES

We consider two techniques: 1) the multi-condition training of the acoustic model; and 2) the removal of background music using a denoising autoencoder and subsequent acoustic model training on the processed data. We consider two types of autoencoders: a fully connected and the convolutional network.

The configuration of hyper-parameters for all acoustic models corresponds to the best performance in preliminary experiments with undistorted data. The configuration for autoencoders was selected based on preliminary experiments with a fully connected network on dataset Piano 1 (see Table 1).

3.1. General acoustic model structure

Apart from the training data, the acoustic models for both approaches are similar, based on Hidden Markov Model - Deep Neural Network (HMM-DNN) hybrid architecture [17]. The underlying Gaussian Mixture Model is trained as context dependent, speaker independent and contains 2219 physical states.

For feature extraction, filter bank coefficients [18] are computed using 25 ms frames of signal with frame shifts of 10 ms. To normalize the features, Cepstral Mean Subtraction ([19], CMS) with a floating window of 1 s is employed. The input for DNN consists of 11 consecutive feature vectors, 5 preceding and 5 following the current frame.

The Torch library [20] is used for the DNN training, which has a fixed duration of 20 epochs. The networks are fully connected and have feed-forward structure with 5 hidden layers. The activation function is ReLU . Each hidden layer consists of 768 units. The mini-batch size is 1024 input vectors and the learning rate is 0.08.

As our *baseline model*, we consider a single-style acoustic model, trained on an undistorted instance of the training dataset.

3.2. Multi-condition training of acoustic model

To train the multi-condition model, we prepare each dataset in the following way. We select N desired SNR levels (details are provided in Table 1). Subsequently, we split the speech corpus into $N + 1$ parts. The first part is left undistorted. To all other parts we add corresponding music, scaled to the predefined average SNR level. The average SNR is computed per one file of speech recordings, which usually corresponds to about two sentences (about 20 words).

We study three different multi-style models, based on Piano and Electronic music in the background of the training speech; details are provided in Table 1. The two piano-based training sets differ in energy levels of the noise; we aim to study influence of unseen noise-intensity conditions. In the experiments, we will denote the multi-condition models by notation MC:Train set, e.g., MC:Piano 1.

3.3. Fully connected feed-forward denoising autoencoder

Our fully connected denoising autoencoder is a feed-forward deep neural network, where all neurons in the lower hidden layer are connected to all neurons in the higher layer. It accepts distorted features at its input layer. The output is an estimate of clean speech features. During the training stage, the autoencoder requires pairs of corrupted and undistorted utterances. In this work, the undistorted data consists of 132 hours of training Czech speech (similar to acoustic model training) and its distorted counterpart is generated artificially, in a manner described in Section 3.2

The autoencoder is trained using the filter bank features (similar to acoustic model training). The training minimizes the mean square distance between the distorted input and the clean target. This criterion is sensitive to scaling, thus we normalize both training and test data (each feature separately) to zero mean and unitary variance. The same normalization values are utilized later in the test phase.

Our autoencoder is constituted of three hidden layers, with 1024 neurons in each layer. We use the ReLU activation function, a learning rate of 0.03 and a batch size of 512 samples. The training is always stopped after 20 epochs.

We denote models trained on the data processed using a fully connected autoencoder by the notation AE:Train set, e.g., AE:Piano 1; the setups are summarized in Table 1.

3.4. Convolutional denoising autoencoder

The convolutional autoencoder represents another network topology, in which the neurons in the higher hidden layer have connections to only several neurons in the lower layer. This model has been

proposed for acoustic modeling and feature extraction in ASR context in [21, 22]. Its advantages over a fully connected network include: easier modeling of translational variance within speech signals, which exist due to different speaking styles [23], and modeling of local correlations within spectral representations of the speech.

We denote models trained on data processed by convolutional autoencoder by the notation CAE:Train set, e.g., CAE:Piano 1; the setups are summarized in Table 1.

The input feature vectors, targets, the training dataset, the activation functions and optimizing criterion remain the same as for the AE. The topology of the two autoencoders differ in two aspects: 1) the input layer; and 2) the substitution of the first hidden layer of the AE by two convolutional layers in CAE (the number of hidden units remains constant).

The input of CAE consists of 11 feature maps, which correspond to 11 following frames in the input feature vector. Each feature map is 39 elements long (number of filter bank features for a single frame). The convolutional kernel in both layers is of size 5×1 (i.e., the weights are shared in frequency only). Between the convolutional layers, there is a max-pooling layer; we use max-pooling by factor of 3. The first hidden layer has 13 feature maps (i.e., 13×39 hidden units) and the second one 39 (i.e., 39×13 hidden units).

4. EXPERIMENTS

We report the results of our experiments via recognition accuracy [%]; all improvements are stated as absolute.

4.1. Description of the test set

We consider two types of data involved in our experiments: 1) The artificially generated data; and 2) the real-world speech recordings with music in the background.

The generated datasets share common test speech recordings. The set has a duration of 2 hours and 44 minutes (13622 words) and it consists of dictated texts, recorded in a silent environment via close-talk microphone. To the speech, we add piano tracks (8 minutes), piano and violin compositions (2 hours and 24 minutes) and electronic music (40 minutes) with various SNR levels. We concatenate the available music as is necessary, to create background for the whole test-speech set. For each scenario with a specific music type and SNR level, we replicate the whole test dataset. Details of the resulting datasets are summarized in Table 2. The piano and violin compositions represent mismatched training-test conditions for all variants of acoustic models. For Test:Electro dataset, the very low SNR levels are omitted, because the scenario is too complicated then (unintelligible even for human listener).

The real-world dataset was created by the authors solely for the purposes of this paper and consists of 17 minutes and 22 seconds of speech (2222 words), recorded from a digital broadcast of a local radio station (Radiožurnál [24]). The speech comes from several summaries, which are given at the beginning of the news program. A track of electronic music is present in the background. We estimate the average SNR level at about 10 dB.

4.2. Employed recognition engine

We use our own ASR system; its core is formed by a one-pass speech decoder performing a time-synchronous Viterbi search.

The linguistic part of the system consists of a lexicon and a language model. We use two types of language models: 1) A model originating from newspaper texts for the scenarios with simulated data; and 2) A model originating from broadcast transcriptions for the scenario with real-world data.

The lexicon contains 550k entries (word forms and multi-word collocations) that were observed most frequently in the corpora covering newspaper texts. The employed Language Model (LM) is based on N-grams. Due to the very large vocabulary size, the system uses bigrams. Our supplementary experiments showed that the bigram structure of the language model results in the best ASR performance with reasonable computational demands.

4.3. Matched training-test conditions

Here, we discuss performance achieved in scenarios with music genres and SNR levels available during training. See Tables 3 and 4, numbers styled in bold italics.

The baseline model achieves recognition accuracy of 85.0% on undistorted data. For this case, the robust models achieve comparable results (degradation by $0.1 - 1.1$ %), i.e., the robustness on distorted data is not achieved at the cost of worse performance on clean speech.

Within the *Test:Piano,* the accuracy of the baseline model deteriorates with increasing amounts of added background music. The decrease is 16.9% for the SNR level at 0 dB. All considered robust techniques achieve much lower degradation ($1.3 - 2.2\%$). Comparing MC models and AE/CAE models, their results are comparable. The performance of more general models Piano 2 (trained on a broader range of SNR levels) is comparable to the more specific Piano 1.

In the *Test:Electro* scenario, the accuracy of the baseline model deteriorates even more noticeably. The decrease is 46.1% for the SNR level at 0 dB. The robust techniques are able to improve this result by up to 35.8%. The model MC:Electronic achieves significantly better results than AE:Electronic and CAE:Electronic, especially at lower SNR levels (9.1% and 12.3% at SNR 0 dB, respectively). We hypothesize that the autoencoders require more training data, when complex multi-instrumental music is considered. Considering higher number of hidden units in AE/CAE for this scenario, a complimentary experiment (omitted due to lack of space) showed some increase in the accuracy, but not up to the levels of MC.

Table 3. Accuracy [%] achieved on the Test:Piano dataset. The numbers styled in bold italics denote matched train-test conditions; normal font denotes unseen SNR levels.

Model	**SNR levels**				
	clean	10	0	-10	-20
Baseline	***85.0***	82.0	68.1	41.4	16.4
MC:Piano 1	***84.9***	***84.5***	***83.5***	77.7	52.3
AE:Piano 1	***84.8***	***84.6***	***83.4***	77.2	52.6
CAE:Piano 1	***84.8***	***84.5***	***83.3***	77.9	54.4
MC:Piano 2	***84.8***	***84.3***	***83.7***	***81.6***	***72.3***
AE:Piano 2	***83.8***	***83.5***	***82.8***	***79.3***	***67.6***
CAE:Piano 2	***83.9***	***83.7***	***82.8***	***80.1***	***70.3***

4.4. Mismatched training-test conditions

This section discusses scenarios, in which the systems were exposed to data with unseen SNR levels (negative SNR in Tables 3 and 4), and unobserved music genres (piano and violin compositions in Table 5).

Within the *Test:Piano,* the accuracy baseline model falls to 16.4%. All Piano 1 models are able to partly improve by up to

Table 4. Accuracy [%] achieved on the Test:Electro dataset. The numbers styled in bold italics denote matched train-test conditions; normal font denotes unseen SNR levels.

Model	SNR levels				
	clean	10	5	0	-5
Baseline	***85.0***	78.9	65.1	38.9	18.4
MC:Electronic	***84.8***	***83.6***	***81.6***	***74.7***	53.1
AE:Electronic	***84.5***	***82.3***	***78.6***	***65.6***	38.4
CAE:Electronic	***84.1***	***81.9***	***77.8***	***62.4***	36.2

38% (SNR level -20 dB). The CAE:Electronic achieves the highest accuracy for very low SNR. The access to data with negative SNR levels (i.e., the Piano 2 models) during training improves the results considerably, improving the baseline performance by up to 55.9%.

In the *Test:Electro* scenario, the baseline model performs poorly, below 18.4% accuracy. Even the robust techniques are only partially able to compensate the difficult acoustic conditions, achieving 53.1% accuracy, for SNR level -5 dB. The MC:Electronic performs substantially better than both autoencoder models (by up to 14.7%). This corroborates the lower performance of autoencoders in more difficult scenarios.

The results achieved on *Test:Violin* demonstrate that the studied techniques are functional on unobserved music genres and improve their accuracy over the baseline recognizer (up to 24.3% at a SNR level of 0 dB). The MC models are more robust with respect to unseen music genres than the AE/CAE models.

Considering the positive SNR levels, the best results are achieved using MC:Electronic model, trained on the broadest spectrum of music genres. This indicates that for the sake of scenarios with mismatched training-test conditions, it is beneficial to include a broad range of genres in the training set. In the case of negative SNR levels, the best performance is achieved by MC:Piano2, which had access to negative SNR levels during training (but not the music genre). This indicates that the benefits of adding broad spectrum SNR levels in the training set are preserved even for unobserved music genres during tests.

Table 5. Accuracy [%] achieved on the Test:Violin dataset. Bold italics denotes matched train-test conditions; normal font denotes unseen music genre and/or SNR levels.

Model	SNR levels				
	clean	10	0	-10	-20
Baseline	***85.0***	76.2	46.8	18.2	5.7
MC:Piano 1	***84.9***	83.0	69.4	41.4	15.6
AE:Piano 1	***84.8***	82.1	64.8	37.2	14.0
CAE:Piano 1	***84.8***	82.0	65.8	37.9	14.3
MC:Piano 2	***84.8***	81.4	68.4	44.2	21.5
AE:Piano 2	***83.8***	80.5	63.9	38.7	16.6
CAE:Piano 2	***86.9***	81.1	66.4	40.9	18.9
MC:Electronic	***84.8***	83.5	71.1	39.0	13.5
AE:Electronic	***84.5***	81.4	62.2	31.7	9.9
CAE:Electronic	***84.1***	80.7	60.5	30.3	9.1

4.5. Real-world test set

The testing on our real-world dataset can be considered to be under mismatched training-test conditions. The included music is of a genre similar (but not identical) to music samples in the Electronic dataset. We estimate the SNR level of these recordings to be about 10 dB. The robust techniques are able to improve over the baseline recognizer by about 2.4%, which corresponds to the improvement in the simpler Test:Piano scenario at SNR level 10 dB.

Table 6. Accuracy [%] achieved on the Real-world dataset (mismatched training-test conditions; unseen music genre).

Model	
Baseline	83.7
MC:Elect 1	86.1
AE:Elect 1	85.8
CAE:Elect 1	86.1

5. CONCLUSIONS AND DISCUSSION

From the results stated above, we draw the following conclusions: 1) Both studied techniques are able to compensate for the performance decrease (caused by interfering music) encountered by a single-style baseline model. 2) The accuracy achieved by both techniques is comparable for matched train-test conditions and simpler background music. 3) The multi-condition models exhibit superior accuracy for mismatched training-test scenarios (with unseen music genre and/or SNR level) and for more complex background music. We hypothesize that more complicated scenarios will require more data to train the autoencoders. This holds for the Electronic training dataset (which consists of a broad spectrum of music genres) and also for the Piano 2 dataset (which contains a broad range of SNR levels). 4) Comparing both autoencoder topologies, the fully connected one achieves a higher performance compared to convolutional one in more difficult scenarios. The convolutional autoencoder exhibits a higher performance for simpler scenarios and lower SNR levels. 5) In accordance with literature, the models trained with broader range of music genres are more robust in mismatched train-test conditions. 6) The access to a broader range of SNR levels during the training helps in scenarios with similar SNR levels and unseen music genres.

The comparison of autoencoders is partly in contrast with the results presented in [15], where the performance of the convolutional autoencoder was superior to a fully connected case for all considered scenarios. We argue that this could be caused by: 1) the lower number of hidden units in our CAE compared to [15] (which we keep equal to number of neurons in AE). Our complimentary experiment confirmed that CAE benefits from the increased number of neurons significantly more than AE. 2) We consider more general target music (in [15], the autoencoders are trained for a specific "song", we train with respect to a general genre).

Concerning computational demands, the multi-condition training is less demanding, requiring training/utilization of only a single network. The advantage of autoencoders dwells in the simplicity of obtaining large amount of data for training, because there is no need for reference texts labeled manually. This fact inspires our future work, in which we will study the size of the datasets required for efficient training of the studied techniques and the benefits of increasing/decreasing that size. We expect that the training of an acoustic model on data preprocessed by the autoencoder requires a smaller (labeled) dataset in comparison with full multi-conditional training. Moreover, the autoencoder can be trained in a multilingual fashion [15], serving as a preprocessing tool for several language-specific acoustic models.

6. REFERENCES

[1] Jinyu Li, Li Deng, Yifan Gong, and Reinhold Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 4, pp. 745–777, 2014.

[2] Michael L Seltzer, Dong Yu, and Yongqiang Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7398–7402.

[3] Keizo Kinoshita, Marc Delcroix, Takashi Yoshioka, Takeshi Nakatani, Armin Sehr, Walter Kellermann, and Roland Maas, "The reverb challenge: Acommon evaluation framework for dereverberation and recognition of reverberant speech," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on*. IEEE, 2013, pp. 1–4.

[4] "Reverb challenge [online]," http://reverb2014.dereverberation.com/, Accessed: 2016-29-08.

[5] Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee, "An experimental study on speech enhancement based on deep neural networks," *Signal Processing Letters, IEEE*, vol. 21, no. 1, pp. 65–68, 2014.

[6] Xugang Lu, Yu Tsao, Shigeki Matsuda, and Chiori Hori, "Speech enhancement based on deep denoising autoencoder.," in *INTERSPEECH*, 2013, pp. 436–440.

[7] Emmanuel Vincent, Jon Barker, Shigetaka Watanabe, Jonathan Le Roux, Francesco Nesta, and Marco Matassoni, "The second chimespeech separation and recognition challenge: Datasets, tasks and baselines," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 126–130.

[8] Shoko Araki, Tomoki Hayashi, Marc Delcroix, Masakiyo Fujimoto, Kazuya Takeda, and Tomohiro Nakatani, "Exploring multi-channel features for denoising-autoencoder-based speech enhancement," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 116–120.

[9] Jon Barker, Ricard Marxer, Emmanuel Vincent, and Shinji Watanabe, "The thirdchime'speech separation and recognition challenge: Dataset, task and baselines," in *2015 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU 2015)*, 2015.

[10] Marc Delcroix, Yotaro Kubo, Tomohiro Nakatani, and Atsushi Nakamura, "Is speech enhancement pre-processing still relevant when using deep neural networks for acoustic modeling?," in *INTERSPEECH*. 2013, pp. 2992–2996, ISCA.

[11] Angkana Chanrungutai and Chotirat Ann Ratanamahatana, "Singing voice separation for mono-channel music using non-negative matrix factorization," in *Advanced Technologies for Communications, 2008. ATC 2008. International Conference on*. IEEE, 2008, pp. 243–246.

[12] Emad M Grais and Hakan Erdogan, "Single channel speech music separation using nonnegative matrix factorization and spectral masks," in *Digital Signal Processing (DSP), 2011 17th International Conference on*. IEEE, 2011, pp. 1–6.

[13] Pablo Sprechmann, Alexander M Bronstein, and Guillermo Sapiro, "Real-time online singing voice separation from monaural recordings using robust low-rank modeling.," in *ISMIR*, 2012, pp. 67–72.

[14] Cemil Demir, Murat Saraclar, and Ali Taylan Cemgil, "Single-channel speech-music separation for robust asr with mixture models," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 4, pp. 725–736, 2013.

[15] Mengyuan Zhao, Dong Wang, Zhiyong Zhang, and Xuewei Zhang, "Music removal by convolutional denoising autoencoder in speech recognition," in *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. IEEE, 2015, pp. 338–341.

[16] "Free music archive [online]," http://freemusicarchive.org/, Accessed: 2016-08-29.

[17] George Dahl, Yu Dong, Li Deng, and Alex Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 30 –42, jan. 2012.

[18] Steve Young, "The htk hidden markov model toolkit: Design and philosophy," *Entropic Cambridge Research Laboratory, Ltd*, vol. 2, pp. 2–44, 1994.

[19] Richard J Mammone, Xiaoyu Zhang, and Ravi P Ramachandran, "Robust speaker recognition: A feature-based approach," *Signal Processing Magazine, IEEE*, vol. 13, no. 5, pp. 58, 1996.

[20] "Torch - a scientific computing framework for luajit [online]," http://torch.ch, Accessed: 2016-08-29.

[21] Tara N Sainath, Abdel-rahman Mohamed, Brian Kingsbury, and Bhuvana Ramabhadran, "Deep convolutional neural networks for lvcsr," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8614–8618.

[22] Yajie Miao and Florian Metze, "Improving language-universal feature extraction with deep maxout and convolutional neural networks," 2014.

[23] Yann LeCun and Yoshua Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, pp. 1995, 1995.

[24] "Cesky rozhlas - radio station radiozurnal [online]," http://www.rozhlas.cz/radiozurnal/, Accessed: 2016-08-29.

ROBUST RECOGNITION OF SPEECH WITH BACKGROUND MUSIC IN ACOUSTICALLY UNDER-RESOURCED SCENARIOS

Jiri Malek, Jindrich Zdansky and Petr Cerva

Faculty of Mechatronics, Informatics, and Interdisciplinary Studies, Technical University of Liberec, Studentská 2, 461 17 Liberec, Czech Republic.

ABSTRACT

This paper addresses the task of Automatic Speech Recognition (ASR) with music in the background. We consider two different situations: 1) scenarios with very small amount of labeled training utterances (duration 1 hour) and 2) scenarios with large amount of labeled training utterances (duration 132 hours). In these situations, we aim to achieve robust recognition. To this end we investigate the following techniques: a) multi-condition training of the acoustic model, b) denoising autoencoders for feature enhancement and c) joint training of both above mentioned techniques.

We demonstrate that the considered methods can be successfully trained with the small amount of labeled acoustic data. We present substantially improved performance compared to acoustic models trained on clean speech. Further, we show a significant increase of accuracy in the under-resourced scenario, when utilizing additional amount of non-labeled data. Here, the non-labeled dataset is used to improve the accuracy of the feature enhancement via autoencoders. Subsequently, the autoencoders are jointly fine-tuned along with the acoustic model using the small amount of labeled utterances.

Index Terms: robust speech recognition, feature enhancement, denoising autoencoder, multi-condition training, joint training.

1. INTRODUCTION

Nowadays, the research in automatic speech recognition (ASR) is focused on robustness of the performance with respect to difficult environmental conditions. An example of such conditions arising naturally in real-world is background noise. The robustness-introducing techniques most often focus on environmental noise, such as street or restaurant sounds [1]. Principally different type of interference is music, which is however less considered in the ASR literature. Yet, it is one of the often encountered background sounds in applications such as online $24/7$ monitoring of broadcast media.

In our recent paper [2], we analyzed two popular approaches to robust ASR in the context of background music. The first approach was the *multi-condition training* (MCT) of acoustic models; we considered Fully-connected deep neural network Acoustic Models (FAM). Here, the model incorporates the knowledge on possible interferences through the inclusion of the distorted signals in the training set. For non-musical environmental noise, this approach was reported to obtain high performance in [3]. Besides, this technique was demonstrated to be beneficial for reverberated speech in [4, 5].

Another analyzed approach is the feature preprocessing using *denoising autoencoders* (AE, [2, 6]). In our context, the denoising autoencoder is a feed-forward deep neural network, either fully-connected (FAE) or a convolutional one (CAE). It aims at separation of the speech features from the interfering music, i.e., the ASR is subsequently performed on the enhanced features. Considering the environmental noise, the benefits of autoencoders for ASR were shown in [7], where the car and factory noises were considered. Another network topology for autoencoders, based on Bidirectional Long Short-Term Memory (BLSTM) recurrent neural networks, was presented in [8]. The front-end preprocessing usually introduces distortions into enhanced data, which are not observed by the acoustic model trained on the clean data. To mitigate, the enhancement is usually applied on both training and test data and the new acoustic model is trained on the enhanced dataset [9].

Relation to prior work: We presented in [2] that both of the above mentioned techniques are able to significantly improve the recognition of speech with background music. When comparing the two, we found the multi-condition training achieving slightly superior results, especially for mismatched training-test conditions and more complex background music.

Unlike the previous work, this paper investigates the suitability of the above-mentioned techniques in a scenario where a very small amount of labeled training speech is available (duration of about 1 hour). This problem can be encountered, e.g., when building a recognizer for a new language or when dealing with an under-resourced language [10]. Since speech labeling is costly and time-consuming, we also investigate the possibility of improving the performance using a larger amount of non-labeled speech. We compare the performance of these under-resourced models to models trained using a large amount of labeled speech.

Next, we extend the portfolio of the considered robust techniques. Taking into account the advantages of convolutional topology reported in [11], we consider the *Convolutional Acoustic Models for the Multi-Condition Training* (MCT-CAM). The convolutional models reflect strong correlations of speech in time and are invariant to translational variance within speech caused, e.g., by different speaking styles. Further, we attempt to combine the benefits of both above-mentioned approaches using the *Joint Training* of the acoustic model and convolutional autoencoder (JMCT) proposed in [12]. This approach fine-tunes both the feature enhancement by CAE and the acoustic model, exploiting the information about senone classification instead of optimizing just the squared error as in CAE.

Finally, we perform a more detailed analysis of the autoencoder performance with respect to its topology than was performed in [2]. There we found that the performance of FAE is comparable to the performance of CAE, assuming both networks have a comparable number of hidden units. This, however, bestows the CAE with a lower number of free parameters. This paper shows that CAE outperforms the FAE, when deeper network and broader convolutional layers are used.

We evaluate the functionality of the methods on artificial mixtures of speech and music, as well as real-world radio shows.

This work was supported by the Technology Agency of the Czech Republic (Project No. TH03010018).

2. PROBLEM FORMULATION AND DATA DESCRIPTION

We focus on the robustness of ASR to music in the background of speech. All of the considered training data are generated artificially, by summation of the speech and music signal. We analyze different scenarios with respect to average Signal-to-Noise Ratio (SNR).

We focus on the *Electronic* music (dataset duration 667 minutes), because it resembles the background music of TV shows. The music originates at the database of free music tracks at the Free Music Archive [13] and consists of genres such as ambient, dance, down-tempo, chillout or IDM.

We consider a Large Vocabulary Continuous Speech Recognition (LVCSR) task. Due to the data most readily available to us, we focus on the Czech language, without any loss of generality to the investigated problems. Our available dataset of clean speech consists of 132 hours of Czech utterances.

We use two different sizes of training datasets throughout the experiments. *The large training dataset* contains all available utterances, i.e., 132 hours of labeled speech. *The small training dataset* is a subset of the former, which contains 1 hour of labeled speech. We use this dataset to study the considered techniques in scenarios similar to under-resourced languages. We select the sentences for the subset, such that all Czech phonemes are sufficiently present to successfully train the acoustic models.

In the context of the small dataset, we investigate one more scenario, where next to the 1 hour of labeled data we also have 20 hours of non-labeled data. Compared to labeling the data, the non-labeled speech is easier to obtain. It cannot be used directly to train the acoustic models, but it can be used to improve the performance of the autoencoders. However, the acoustic models can also benefit from the enlarged amount of data due to joint training/fine-tuning with the autoencoders.

3. PROPOSED ROBUSTNESS-INTRODUCING TECHNIQUES

We consider three techniques: 1) The multi-condition training (MCT) of the acoustic model, either a fully-connected (FAM) or convolutional (CAM) one. 2) The denoising autoencoder trained to remove the background music from the features and subsequent FAM training on the processed data. For this we utilize two types of autoencoder: the fully-connected (FAE) and the convolutional network (CAE). 3) The joint multi-condition training of CAE and FAM using noisy training data (JMCT).

The configuration of hyper-parameters for all acoustic models corresponds to the best performance in preliminary experiments with undistorted data. The configuration for autoencoders was selected based on experiments in Section 4.3.

All neural networks are trained using the Torch library [14]. The training procedure ends when the respective optimization criterion does not improve anymore on a small validation dataset, which is not part of the training set. We use the ReLU activation function within the networks.

For feature extraction, 39 filter bank coefficients [15] are computed using 25-ms frames of signal and frame shift of 10 ms. The input for DNNs consists of 11 consecutive feature vectors, 5 preceding and 5 following the current frame.

3.1. General acoustic model structure

The FAM/CAM networks trained by MCT and on data produced by autoencoders share many common topological features and hyper-parameters. All models are based on Hidden Markov Model-Deep Neural Network (HMM-DNN) hybrid architecture [16]. The underlying Gaussian Mixture Model (GMM) is trained as context dependent, speaker independent.

We use the two above-mentioned sizes of training set, i.e., 1 hour and 132 hours. The GMM model corresponding to the small dataset contains 619 physical states. The underlying GMM model for the large dataset contains 2219 physical states.

The acoustic models are trained using minimization of the negative log-likelihood criterion. As feature normalization, we employ the Mean Subtraction ([17]) with a floating window of 1 s.

As our *baseline acoustic model*, we consider a single-style model (SCT). It shares the topology described above and is trained on an undistorted instance of each training dataset.

3.2. Multi-condition training of acoustic model

To train the multi-condition model, we prepare each dataset in the following way. We select three desired SNR levels (10, 5 and 0 dB). Subsequently, we split the speech corpus into four parts. The first part is left undistorted. To all other parts we add corresponding music, scaled to the predefined average SNR level. The average SNR is computed per one file of speech recordings, which usually corresponds to about two sentences (about 20 words).

The FAMs have a feed-forward structure with five fully-connected hidden layers. Each hidden layer consists of 768 units.

The CAMs are comprised of two convolutional layers and three fully connected layers (consisting of 768 units). The input consists of 11 feature maps, each 39×1 in size, which correspond to the 11 consecutive feature vectors. Based on experiments with autoencoder topology from Section 4.3, the first layer consists of 105 feature maps 39×1 in size, and the second layer of 157 feature maps 13×1 in size. There is a $3:1$ max-pooling layer situated between the convolutional layers.

3.3. Fully-connected feed-forward denoising autoencoder

Our FAE is a feed-forward deep neural network, where all neurons in the lower hidden layer are connected to all neurons in the higher layer. It accepts distorted features at its input layer. The output is an estimate of clean speech features. During the training, the autoencoder requires pairs of corrupted and undistorted speech. Our undistorted data consists of Czech speech with datasets similar to the ones used for MCT. The distorted counterpart is generated artificially as described in Section 3.2.

The network is trained to minimize the mean square distance between the distorted input and the clean target. This criterion function is sensitive to scaling, thus we normalize both training and test data (each feature separately) to zero mean and unitary variance.

Our autoencoder is constituted of three or four hidden layers (see Section 4.3), with 1024 neurons in each layer. We use the ReLU activation function.

3.4. Convolutional denoising autoencoder

The CAE represents another network topology, in which the neurons in the higher hidden layer have connections to only several neurons in the lower layer. This model has been proposed for acoustic modeling and feature extraction in ASR context in [18, 19].

The input feature vectors, targets, the training dataset, the activation functions, and the optimizing criterion remain the same as for the FAE. The topology of the two autoencoders differ in two aspects:

1) the input layer; and 2) the replacement of the first two hidden layers of the FAE with two convolutional layers in CAE (see Section 4.3 for details).

The input of CAE consists of 11 feature maps, which correspond to 11 following frames in the input feature vector. Each feature map is 39 elements long (the number of filter bank features for a single frame). The convolutional kernel in both layers is of size 5×1 (i.e., the weights are shared in frequency only, as suggested in [19]). Between the convolutional layers, there is a max-pooling layer; we use max-pooling by a factor of 3.

3.5. Joint training of CAE and FAM

We perform the joint training (JMCT) in the following manner, similar to paper [12]. 1) The CAE is trained as is described in Section 3.4, with the following two exceptions: a) We use as targets eleven consecutive frames of clean speech, not only the current single frame as in Section 3.4; and b) the CAE contains only a single fully-connected hidden layer consisting of 768 units. 2) We train a FAM network using the data processed by this CAE, i.e., the input vector consists of 39×11 features for each speech frame. The FAM model contains two hidden layers with 768 units. 3) We directly stack the acoustic modeling layers on top of the autoencoder layers, which means that the output layer of the autoencoder is similar to the input layer of the acoustic model. 4) All weights in the resulting network are fine-tuned using the negative log-likelihood criterion. The convolutional acoustic model resulting from the joint training is thus similar in size and topology to the MCT-CAM model described in Section 3.2.

During the joint training, we encountered very slow convergence speed after stacking the CAE and FAM. To mitigate this phenomenon, we apply batch normalization [20] of hidden layers within the FAM network in step 2).

4. EXPERIMENTS

We report the results of our experiments via recognition accuracy [%]; all improvements are stated as absolute.

4.1. Description of the test set

We consider two types of test data in our experiments: 1) The artificially generated data; and 2) the real-world speech recordings with music in the background.

The generated dataset has a duration of 2 hours and 44 minutes (13622 words) and consists of texts dictated in a silent environment via a close-talk microphone. To the clean speech we add the electronic music (40 minutes) with four distinct SNR levels; 10 dB, 5 dB, 0 dB and -5 dB. We concatenate the available music as is necessary, to create background for the whole test-speech set. We replicate the whole test dataset for each scenario with a specific SNR level.

The real-world dataset consists of 17 minutes and 22 seconds of speech (2222 words), recorded from a digital broadcast of a local radio station (Radiožurnál [21]). The speech comes from several summaries, which are given at the beginning of the news program. A track of electronic music is present in the background. We estimate the average SNR of the dataset at about 10 dB using method from [22].

4.2. Employed recognition engine

We use our own ASR system; its core is formed by a one-pass speech decoder performing a time-synchronous Viterbi search.

The linguistic part of the system consists of a lexicon and a language model. In this paper, we assume that there is a sufficient amount of linguistic data to create a functional model, i.e., we do not investigate the under-resourced scenario from the linguistic point of view. We use two types of language models: 1) A model originating from newspaper texts for the scenarios with the simulated data; and 2) A model originating from broadcast transcriptions for the scenario with real-world data.

The lexicon contains 550k entries (word forms and multi-word collocations) that were observed most frequently in the corpora covering newspaper texts. The employed Language Model (LM) is based on N-grams. Due to the very large vocabulary size, the system uses bigrams. Our supplementary experiments showed that the bigram structure of the language model results in the best ASR performance with reasonable computational demands.

4.3. Comparison of the autoencoder topologies

In this section, we supplement the comparison of the autoencoders presented in our previous paper [2] and perform a hyper-parameter selection for FAE and CAE. The best configurations (FAE-2 and CAE-4) are used further in Sections 4.4 and 4.5.

The comparison in [2] was based on an equal number of hidden units/layers of the respective networks (FAE-1 and CAE-1). This is somewhat unfair for the CAE, it forces it to have a smaller number of free parameters. In Tables 1 and 2, we present a more balanced analysis, observing the number of free parameters within the models, as was presented, e.g., in [11]. The autoencoders were trained using the large training dataset.

Table 1. Accuracy[%] achieved by autoencoder enhancement on the real-world dataset. Column Maps describes numbers of feature maps in the first and second convolutional layers, respectively. Bold numbers indicate the highest accuracy.

Method	Layers	Params	Maps	Accuracy[%]
FAE-1	3	2.6M	0/0	85.8
FAE-2	4	3.6M	0/0	85.2
CAE-1	3	1.6M	13/39	**86.1**
CAE-2	4	2.1M	26/78	85.3
CAE-3	4	2.2M	52/78	85.6
CAE-4	4	3.3M	105/157	85.0

Table 2. Accuracy[%] achieved by autoencoder enhancement on the generated dataset with respect to average SNR level. Bold numbers indicate the highest accuracy for given SNR level.

Method	Params	**SNR-level**				
		Clean	10dB	5dB	0dB	-5dB
FAE-1	2.6M	84.4	82.3	78.6	65.4	38.3
FAE-2	3.6M	**84.5**	82.1	78.7	66.6	39.9
CAE-1	1.6M	84.0	81.8	77.7	62.2	36.0
CAE-2	2.1M	84.3	82.2	79.0	66.8	40.3
CAE-3	2.2M	**84.5**	82.5	79.6	69.0	43.5
CAE-4	3.3M	84.4	**82.7**	**79.7**	**69.7**	**44.3**

The results, presented in Tables 1 and 2, indicate that the FAEs do not benefit much from increasing the number of free parameters from 2.6M to 3.6M. Considering the comparable number of free parameters (FAE-2 and CAE-4), the CAE outperforms the FAE. This

is in concert with the literature describing the convolutional autoencoders [6]. The CAE emphasizes more than the FAE the strong dependence between features, which are close in time and frequency. With respect to the comparison in [2], the increased performance of CAE is caused: 1) as expected, by an increased number of free parameters (see CAE-1 and CAE-2) and 2) by utilization of a broader first layer, especially in experiments considering lower SNR levels (see CAE-2 and CAE-3).

Considering the real dataset (where we estimate the SNR level at about 10 dB), the differences between the AEs diminish. This result is consistent with the results achieved on the generated dataset and high SNR levels.

4.4. Evaluation of models trained on the small dataset

In Table 3 we analyze the behavior of the models trained on the small dataset (1 hour duration).

All considered techniques improve the performance over the baseline SCT acoustic model. The least effective in this context appears to be the standalone utilization of the autoencoders. The CAE is superior to FAE, but does not achieve the performance of the MCT. Utilization of additional non-labeled data improves the performance of autoencoders, e.g., CAE accuracy improves by more than 7 % for SNR level 0 dB.

Investigating the multi-condition training, the MCT-CAM model performs better compared to MCT-FAM, which in concert with the literature [11]. This advantage vanishes for very low SNR. We presume that training CAM using random initialization might be problematic using very small datasets.

We achieved the best results using the joint training. The JMCT models, which are comparable in topology and size to models MCT-CAM, exhibit the higher accuracy of the two. This holds even for very low SNR; here, the performance drop as compared to MCT-FAM does not appear.

Additional hours of non-labeled data are able to improve the recognition accuracy considerably. This corresponds to our scenario when the CAE within JMCT model is trained on 20 hours of non-labeled data and the whole concatenated model is fine-tuned on the 1 hour of labeled data. The JMCT(20h) model consistently outperforms the JMCT(1h) model by $1-4\%$. The pretrained JMCT(20h) model outperforms even the SCT model on clean speech; for which the SCT model is specifically trained.

Considering the real-world dataset, the accuracy the improvements are less significant than on the corresponding augmented dataset with SNR of 10dB. We conjecture that this is caused by the smaller deterioration of the SCT performance on the real dataset.

4.5. Evaluation of models trained on the large dataset

The additional data in the large dataset bring higher accuracy and more robustness to all trained models, as is indicated in Table 4. For example, comparing the SCT model to the under-resourced SCT model, the achieved accuracy is higher by about 8% when transcribing the clean data and by about 16% when recognizing the real-world dataset. Moreover, with decreasing SNR, the accuracy of all models trained on large dataset deteriorates much less rapidly.

The autoencoders achieve comparable results to MCT for a SNR level of 10 dB and higher, unlike to the under-resourced scenario. On lower SNR levels, the MCT outperforms the autoencoders considerably (by more than 7% for SNR 0 dB) .

The MCT-CAM appears to be superior to MCT-FAM for all test datasets. Its accuracy is not even deteriorated on clean data compared to SCT. In contrast to results observed on models trained using the small training set, the joint training improves the performance over MCT-CAM only slightly ($0-1.1\%$).

Investigating the real-world dataset, all the robust techniques are able to improve the results obtained by SCT (by up to 2.7%). The best results are achieved by MCT-CAM and JMCT, which is consistent with results achieved on simulated data.

Table 3. Training set: 1 hour; Accuracy[%] on the generated/real-world dataset. Bold numbers indicate the highest achieved accuracy. The numbers in parentheses describe the amount of non-labeled data to train the autoencoder.

Method	SNR-level (generated)					Real
	Clean	10dB	5dB	0dB	−5dB	
SCT	76.8	59.4	39.8	20.5	10.8	67.5
MCT-FAM	74.9	71.3	61.6	43.5	21.5	69.1
MCT-CAM	76.4	72.1	62.0	40.9	19.5	69.6
FAE(1h)	65.1	51.8	37.9	21.0	11.3	58.0
FAE(20h)	72.8	65.5	54.1	35.8	18.6	66.3
CAE(1h)	71.8	64.5	53.5	34.5	17.1	63.9
CAE(20h)	74.3	68.6	59.4	42.8	23.2	70.8
JMCT(1h)	76.1	72.3	65.1	47.9	24.7	66.9
JMCT(20h)	**77.5**	**73.7**	**67.0**	**52.1**	**27.0**	**70.9**

Table 4. Training set: 132 hours; Accuracy[%] on the generated/real-world dataset with respect to average SNR level. Bold numbers indicate the highest achieved accuracy.

Method	SNR-level (generated)					Real
	Clean	10dB	5dB	0dB	−5dB	
SCT	84.9	78.8	64.8	38.7	18.2	83.7
MCT-FAM	84.7	83.6	81.5	74.5	53.0	86.1
MCT-CAM	84.9	84.0	81.9	76.1	56.5	**86.4**
FAE	84.5	82.1	78.7	66.6	39.9	85.2
CAE	84.4	82.7	79.7	69.7	44.3	85.0
JMCT	**85.1**	**84.2**	**82.3**	**76.7**	**57.7**	**86.4**

5. CONCLUSIONS

From the above-stated results we draw the following conclusions, which hold regardless of the size of the training set. 1) All considered robust ASR techniques are able to improve the results of the SCT baseline model when recognizing speech with background music. 2) Comparing the two autoencoder topologies, the CAE is more suitable for noisy feature enhancement. 3) Comparing the two types of MCT acoustic models, the convolutional one is superior. 4) The best results are achieved using the joint training of autoencoder and acoustic model. This holds even when comparing MCT-CAM and JMCT, which share similar topology and size. This means that a pretrained CAE is suitable as initial layers of the final acoustic model, when it is fine-tuned along with the weights of the acoustic model.

The following conclusions stem from the experiments using models trained on the small dataset. 5) As expected, models trained using the smaller dataset exhibit lesser accuracy and are less robust to background music. 6) An additional amount of *non-labeled* data can considerably improve the performance of any autoencoder type, and can also considerably boost the performance of JMCT systems. This improvement thus brings the benefits of a larger training dataset without the need for any additional labeling of data.

6. REFERENCES

[1] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech & Language*, 2016.

[2] J. Malek, J. Zdansky, and P. Cerva, "Robust automatic recognition of speech with background music," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, vol. 1. IEEE, 2017.

[3] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7398–7402.

[4] K. Kinoshita, M. Delcroix, S. Gannot, E. A. Habets, R. Haeb-Umbach, W. Kellermann, V. Leutnant, R. Maas, T. Nakatani, B. Raj *et al.*, "A summary of the reverb challenge: state-of-the-art and remaining challenges in reverberant speech processing research," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, p. 7, 2016.

[5] Reverb challenge [online]. http://reverb2014.dereverberation.com/. Accessed: 02.10.2017.

[6] M. Zhao, D. Wang, Z. Zhang, and X. Zhang, "Music removal by convolutional denoising autoencoder in speech recognition," in *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. IEEE, 2015, pp. 338–341.

[7] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder." in *INTERSPEECH*, 2013, pp. 436–440.

[8] A. El-Desoky Mousa, E. Marchi, and B. Schuller, "The icstm+ tum+ up approach to the 3rd chime challenge: Single-channel lstm speech enhancement with multi-channel correlation shaping dereverberation and lstm language models," *arXiv preprint arXiv:1510.00268*, 2015.

[9] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third-chime'speech separation and recognition challenge: Dataset, task and baselines," in *2015 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU 2015)*, 2015.

[10] L. Besacier, E. Barnard, A. Karpov, and T. Schultz, "Automatic speech recognition for under-resourced languages: A survey," *Speech Communication*, vol. 56, pp. 85–100, 2014.

[11] T. N. Sainath, B. Kingsbury, G. Saon, H. Soltau, A.-r. Mohamed, G. Dahl, and B. Ramabhadran, "Deep convolutional neural networks for large-scale speech tasks," *Neural Networks*, vol. 64, pp. 39–48, 2015.

[12] T. Gao, J. Du, L.-R. Dai, and C.-H. Lee, "Joint training of front-end and back-end deep neural networks for robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4375–4379.

[13] Free music archive [online]. http://freemusicarchive.org/. Accessed: 02.10.2017.

[14] Torch - a scientific computing framework for luajit [online]. http://torch.ch. Accessed: 02.10.2017.

[15] S. Young and S. Young, "The htk hidden markov model toolkit: Design and philosophy," *Entropic Cambridge Research Laboratory, Ltd*, vol. 2, pp. 2–44, 1994.

[16] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 30 –42, jan. 2012.

[17] R. J. Mammone, X. Zhang, and R. P. Ramachandran, "Robust speaker recognition: A feature-based approach," *Signal Processing Magazine, IEEE*, vol. 13, no. 5, p. 58, 1996.

[18] T. N. Sainath, A.-r. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for lvcsr," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8614–8618.

[19] Y. Miao and F. Metze, "Improving language-universal feature extraction with deep maxout and convolutional neural networks," 2014.

[20] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[21] Cesky rozhlas - radio station radiozurnal [online]. http://www.rozhlas.cz/radiozurnal/. Accessed: 02.10.2017.

[22] C. Kim and R. M. Stern, "Robust signal-to-noise ratio estimation based on waveform amplitude distribution analysis," in *Ninth Annual Conference of the International Speech Communication Association*, 2008.

5 Dealing with multilingual audio data

5.1 Efficient adaptation of an ASR system to a new language

As described in Sec. 2.2.4, when adapting an existing ASR system to a new language, the major development issues and costs are associated with the creation of an appropriate AM. We have therefore developed an approach that allows us to extract speech and prepare suitable training data from the public Internet sources, such as TV and radio recordings [35], parliament archives [36], or recently also YouTube videos. Many of these sources are accompanied by texts in forms of, e.g., subtitles, news briefs, extracted citations, or stenograms (in the case of the parliament meetings). The main advantage of these audio/video sources is that they contain real talks recorded in authentic environments, in contrast to read speech passages often occurring in many of the commercially available speech training databases. However, the problem starts if we want to collect and utilize the mentioned Internet data in a fully automatic manner: a priori, we do not know the relationship between the audio and the available text. It can be rather straightforward, though not exactly verbatim (in the case of the parliament stenograms), it can have a form of a shortened and approximate message (in the case of subtitles), or it can be just an interpretation or summary what has been said.

The article included in this thesis [37], describes the methods and procedures we have developed in order to automate the collection, phonetic transcription and AM training for new languages from public sources. The proposed scheme is iterative and it can be applied both at an initial phase when no AM is available for the target language, as well as in situations when we want to update and improve an existing AM.

In the former case we use a so called bootstrapping approach, where an AM from another (related) language is employed to collect first batches of transcribed and annotated audio. During the bootstrapping phase, we need to create a temporary lexicon with pronunciations represented by the phonetic inventory of the donor language, and the AM is trained on the mix of speech data from the donor and target languages. When a sufficient amount of the speech data from the target language is collected, we can return to the original pronunciations and let the AM iteratively improve on the newly collected data. The whole process is controlled by the ASR system that produces orthographic and phonetic transcriptions, compares them with the accompanying texts, searches for parts of texts and audio where the

ASR output and reference text matches, and adds these matching segments to the training database for the next iteration step. The scheme is rather robust and safe, and it is able to detect potential mismatches even in commercial speech data-sets, as was shown, e.g., in [38].

This scheme was further enhanced to allow for collecting speech data for which no reference texts existed, in an unsupervised mode, see [39]. Here, several ASR systems, each with a different configuration of the AM and LM, are employed to transcribe a speech record. When they produce the same output, we can consider that transcription correct and ready to be added to the training set.

So far, the described approach has been successfully utilized in the AM development for more than 15 languages. I have contributed to its design and implementation namely by automating the training process. Moreover, I have participated in the creation of AMs for Slovak, Polish and Croatian. Recently, we have been employing and further improving the scheme in our project focused on Scandinavian languages.

It should also be noted that the speech data in multiple languages allows us to train a multilingual AM. The current version of this model is based on the FSMN topology and utilizes a block soft-max during training [40, 41]. In this case, a specific output (last) layer of the neural network corresponding to the target language is activated for every input frame during the training while the weights in remaining network layers are trained as language independent using all available data. The main advantage of this approach is that the multilingual layers integrate the acoustic variations for different languages, acoustic channels, or even for specific tasks. The multilingual AM thus yields lower WER in particular for languages with limited amount of available training data. This fact also further helps in the development phase described above.

5.2 Batch and frame-wise language identification

The next three articles are devoted to language identification from an acoustic signal in off-line as well as in frame-wise mode.

Batch mode

The first paper [42] deals with the batch scenario. In this work, various feed-forward fully connected, convolutional and recurrent DNN architectures are adopted and compared against a baseline i-vector based system. Moreover, feed forward fully-connected DNN is also utilized for the extraction of monolingual bottleneck (BTN) features [43] from the input signal.

The experimental evaluation is performed on a data set that is made public for general use[1]. It contains 11×500 utterances belonging to languages that are all related to each other and sometimes hard to distinguish even for human listeners: it is compiled from recordings of the 11 most widespread Slavic languages. These

[1] https://owncloud.cesnet.cz/index.php/s/gXHKFs9UDEqe34G

languages have not been chosen randomly; they represent the target sub-set for our ASR platform. Note that the recordings from this evaluation set are not longer than 5 seconds so that the resulting approach may be utilized even for short utterances.

The obtained results show the BTN features are beneficial for all of the investigated DNN architectures. Without BTNs, the baseline i-vector based system is able to outperform systems with fully connected DNNs as well as convolutional networks. The best results are yielded by a bidirectional recurrent DNN with gated recurrent units that is fed by BTNs. By using the best method, the baseline error rate (ER) of the i-vector based classifier is reduced from 4.2% to 1.2%.

The more detailed analysis of these results has further shown that the worst results have, in most cases, been obtained for pairs of languages that are related to each other and belong to the same branch of Slavic languages; they are thus also difficult to be distinguished by humans. The most challenging pair for identification is Belorussian and Ukrainian (East branch). Other difficult groups of languages to distinguish are Czech, Slovak and Polish (West branch) and Serbian, Croatian and Slovene (from the South branch).

The second paper [44], published within the NordTrans project, deals again with the off-line identification. It focuses on the three main Scandinavian languages (Swedish, Danish and Norwegian), that are closely related to each other similar to languages from individual Slavic branches. Within this work, various state-of-the-art LID approaches have been adopted, compared and combined, including i-vectors, DNNs, two multilingual bottleneck extractors as well as x-vectors.

The best two resulting approaches takes advantage of multilingual BTNs, which are extracted with the aid of vectorized FSMN. This FSMN-based extractor is trained for 17 languages (including, e.g., Swedish and Norwegian, English and 11 main Slavic languages) using a block soft-max. Moreover, 240 h of augmented speech data are added to the training set to improve the robustness to a) reverberation/noise [45] and b) telephone/speech codecs [11].

These BTN features form an input into a) the classifier based on time-delay neural networks (TDNN) [46] or b) an FSMN-based x-vector extractor, whose output is classified by using a fully connected DNN-based classifier with only two hidden layers. Both these architectures have similar computation demands and they yield comparable ER values of around 1.0%, which makes them suitable for all applications where multiple Scandinavian languages may occur.

Within identification of Norwegian, we also focus on an unexplored sub-task of distinguishing between its two standards – Bokmål and Nynorsk. We assume that acoustic differentiation between them is possible, albeit only to a limited extent, on the basis of their origin and other slight differences in spelling and pronunciation. We focus on this sub-tasks since the existence of these two variants significantly complicates ASR of Norwegian. The phenomenon of their mixing occurs even more often than for individual Scandinavian languages: each time Norwegian is transcribed, one of the standards must be chosen based on the prevailing features in the spoken content, or sometimes also on the speaker's preference. Our results show that distinguishing between Bokmål and Nynorsk from acoustic data is a very hard task: we achieved high ER value of around 20%.

Frame-wise processing

Finally, in the journal article [47], our attention has moved from batch mode to the unexplored frame-wise LID scenario. In this work, the target languages used for evaluation are the Slavic ones. At first, we thus extend the previous off-line evaluation on the data-set of 11 Slavic languages by adopting TDNN and FSMN architectures; these topologies moddel long-term time dependencies in a non-recursive way. We used them not only for classification, but also for extraction of monolingual as well as multilingual and augmented BTN features (similar to the work for Scandinavian languages). In this task, the best resulting approach also corresponds to the TDNN-based classifier, which is fed by multilingual and augmented BTNs extracted by using FSMN. This method yields very a small ER value of merely 0.2%. The detailed results in the form of confusion matrices are depicted in Fig 5.2.

	CZ	SK	PL	RU	SI	UA	RS	MK	HR	BY	BG
CZ	472	10	4	3	1	0	0	0	9	1	0
SK	5	483	5	0	0	0	3	1	0	3	0
PL	9	5	478	5	0	0	0	0	1	1	1
RU	5	3	9	470	1	2	0	0	8	1	1
SI	0	1	0	0	479	4	11	0	2	1	2
UA	1	0	1	1	0	481	0	0	3	9	4
RS	0	3	0	0	10	2	477	1	5	0	2
MK	0	1	0	0	0	2	4	489	1	0	3
HR	8	1	2	5	1	0	6	0	476	0	1
BY	0	1	1	0	4	15	0	2	2	471	4
BG	0	0	0	0	3	0	2	2	0	0	493

	CZ	SK	PL	RU	SI	UA	RS	MK	HR	BY	BG
CZ	497	3	0	0	0	0	0	0	0	0	0
SK	1	499	0	0	0	0	0	0	0	0	0
PL	0	0	500	0	0	0	0	0	0	0	0
RU	0	0	0	500	0	0	0	0	0	0	0
SI	0	0	0	0	500	0	0	0	0	0	0
UA	0	0	0	0	0	499	0	0	0	1	0
RS	0	1	0	0	0	0	497	0	2	0	0
MK	0	0	0	0	0	0	3	497	0	0	0
HR	0	0	0	0	0	0	0	0	500	0	0
BY	0	0	0	0	0	0	0	0	0	500	0
BG	0	0	0	0	0	0	0	0	0	0	500

Figure 5.1: Confusion matrices produced by the baseline i-vector based system (on the left) and the best performing TDNN classifier with FSMN-based BTN features (on the right).

All the findings from off-line evaluation are then utilized for the development of a frame-wise LID approach. This development is performed within a series of consecutive experiments using 7 hours of monolingual Czech and Slovak broadcast programs and 3 hours of bilingual recordings of a talk-show from a Slovak TV channel. The bilingual part of the data comprises ten recordings containing utterances belonging to a Slovak presenter of the show and to 10 different Czech guests of the shows.

The resulting approach is similar to the one used for frame-wise SAD: it takes in a multilingual stream of speech frames and outputs a stream of the corresponding language labels. It employs a weighted finite-state transducer with a proper transduction model as a decoder, which is used for smoothing the output from the TDNN-based language classifier fed by the best FSMN-based BTN features.

The training process of the FSMN-based extractor utilizes 1850 hours of speech data for 11 Slavic languages, 240 hours of augmented data (as in the previous work) and a block soft-max rather than phoneme grouping. The TDNN-based language classifier is trained using three different data-sets. The first artificial one is created

to simulate transitions between the two target languages: it comprises 20 hours of Czech and 20 hours of Slovak utterances that are shuffled and joined in a random order. Given the fact that the average length of an utterance is a few seconds, and there are 40 hours of training data available, a sufficient number of transitions between both languages is ensured. This also includes joined segments in which the language remains unchanged, but the speaker or acoustic channel differs. The second data sub-set contains 80 hours of monolingual Czech and Slovak broadcast programs. Finally, the third part comprises 20 hours of monolingual recordings of the target talk-shows and interviews.

On the bilingual development set mentioned above, our method is capable of detecting language change-points with an F-value of 72.5% and FER of 4.4%. At the same time, it yields FER of 0.4% and 0.8% on the Czech and Slovak monolingual data, respectively. These values prove that it is capable of continuous deployment on broadcast data streams since it does not degrade the results for monolingual programs.

The final evaluation in the frame-wise mode is performed on an independent test set. It comprises ten recordings of interviews from a Czech TV channel. Their total length is 3.25 hours and they contain utterances belonging to a) three different Czech presenters of the program, and b) ten different Slovak guests. Note that all this test data (as well as the development set) is also publicly available together with the data for off-line evaluation.

The obtained results show that the proposed frame-wise LID module allows us to determine the language spoken in these real bilingual TV shows in real-time and with an average latency of around 2.5 seconds At the same time, it increases WER of our ASR system by a mere 2.9% over the reference 18.1% value achieved by using reference language labels prepared by human annotators.

5.3 Reprints

[37] J. Nouza, P. Cerva and M. Kucharova. "Cost-efficient development of acoustic models for speech recognition of related languages." In: *Radioengineering 22.3*, pp. 866–873.

[42] L. Mateju, P. Cerva, J. Zdansky and R. Safarik. "Using Deep Neural Networks for Identification of Slavic Languages from Acoustic Signal." In: *19th Annual Conference of the International Speech Communication Association, INTERSPEECH, Hyderabad, India, 2018*, pp. 1803–1807.

[44] P. Cerva, L. Mateju, F. Kynych, J. Zdansky and J. Nouza. "Identification of Scandinavian Languages from Speech Using Bottleneck Features and X-vectors." In: *24th International Conference on Text, Speech, and Dialogue (TSD), 2021*, Accepted to.

[47] P. Cerva, L. Mateju, J. Zdansky, R. Safarik and J. Nouza. "Identification of related languages from spoken data: Moving from off-line to on-line scenario." In: *Computer Speech and Language 68, 2021.*

Cost-Efficient Development of Acoustic Models for Speech Recognition of Related Languages

Jan NOUZA, Petr ČERVA, Michaela KUCHAŘOVÁ

SpeechLab, Faculty of Mechatronics, Technical University of Liberec, Studentská 2, 46117 Liberec, Czech Republic

jan.nouza@tul.cz, petr.cerva@tul.cz, michaela.kucharova1@tul.cz

Abstract. *When adapting an existing speech recognition system to a new language, major development costs are associated with the creation of an appropriate acoustic model (AM). For its training, a certain amount of recorded and annotated speech is required. In this paper, we show that not only the annotation process, but also the process of speech acquisition can be automated to minimize the need of human and expert work. We demonstrate the proposed methodology on Croatian language, for which the target AM has been built via cross-lingual adaptation of a Czech AM in 2 ways: a) using the commercially available GlobalPhone database, and b) by automatic speech data mining from HRT radio archive. The latter approach is cost-free, yet it yields comparable or better results in experiments conducted on 3 Croatian test sets.*

Keywords

Speech recognition, acoustic model, cross-lingual adaptation, Slavic languages.

1. Introduction

Modern systems for large-vocabulary continuous speech recognition (LVCSR) are designed in the way that allows for easy separation of language dependent and language independent components. The former include an acoustic model (AM), a lexicon with pronunciations, a language model (LM) and an optional text pre-processing and post-processing module. The latter part consists namely of a signal processing front-end and a decoder. When an existing system is to be ported to another language, only the former have to be developed. Thus, there is a natural demand to make this porting in a fast and cost-efficient manner.

As an example, we can mention the efforts of the LIMSI team to adapt their LVCSR system (developed originally for French and English [1]) to other major languages, like e.g. Arabian [2], as well as to those spoken by much smaller population, like Finish [3] or even Luxembourgish [4]. Another research team with a strong focus on multi-lingual speech processing works at the Karlsruhe Institute of Technology. It has collected a large database of spoken data in 20 languages known as GlobalPhone [11] and used it for the development of multi-lingual LVCSR systems. While the creation of the lexicon and the corresponding LM for the target (European) language is the easier task - thanks to digital text resources available via Internet ([5], [6]) - the development of the proper AM requires a large amount of speech records and their phonetic transcriptions. The latter can be provided either manually by an expert (a phonetician) or by automated procedures combined with a varying degree of human supervision [7].

Our research in this field has been motivated by the fact that during the last decade we developed an LVCSR system for Czech that proved to be practically usable in off-line as well as on-line applications, such as broadcast news (BN) transcription [8], spoken archive processing [9] or voice dictation. Later, the system was adapted also to Slovak [10], and recently, we are working on other related languages, like Polish, Russian and Croatian. Our focus on Slavic languages has several rational reasons: a) we can utilize the existing LVCSR system tailored specifically for inflected languages with very large vocabularies, b) we can benefit from the fact that these related languages share some similar and specific patterns in phonetics, lexical inventories, morphology and grammar, c) these languages have attracted less interest from the world-wide research community, so far.

In this paper, we describe the methodology that helped us in a rapid and cost-efficient development of AMs for these languages. In the next section, we provide a brief review of main approaches used for cross-lingual AM adaptation. After that, we propose two methods that reduce the amount of human work in the process of acquisition of phonetically annotated speech data and that can be applied without an expert familiar with the target language. The methods are evaluated experimentally on Croatian, which has been the most challenging language from the above mentioned ones, namely due to limited text and speech resources (as Croatian is spoken by some 5 million people). Yet, the results obtained on 3 different test sets show that the AM created during a several-week period of mostly automated work is applicable for demonstrating a potential of the Croatian LVCSR system.

2. Related Work

Before starting the training of an AM for an LVCSR system, a certain amount of speech from various speakers must be collected and annotated on the phonetic level. For some languages, annotated speech databases suitable for AM training are available, usually on a commercial base. If this is not the case, the phonetic transcriptions must be created, either manually by skilled annotators or by some automated procedures. The well-known Forced Alignment algorithm is the best option when precise orthographic transcriptions are available. In case of large multi-lingual databases, like e.g. GlobalPhone [11], phonetic annotations are missing and the orthographic ones may contain various errors or inconsistencies: from completely or partially wrong texts or corrupted audio, to minor mistakes, like omitted, added or switched words. In such a situation, the transcription process must include a procedure that is capable of discovering and handling these errors. It is usually done by incorporating the iteratively evolving LVCSR system as a checking tool [12].

When an AM for a new language is developed within a multi-lingual environment, the process generally starts by a bootstrapping phase where either one or more existing AMs serve for initializing HMMs of phonemes and noises. The initial model is used to transcribe the data in the target language, which is followed by a series of iterative re-training steps with gradually increasing amount of data. The maximum-likelihood training approach is usually combined with model and feature adaptation techniques as shown, e.g. in [13]. One of the most recent methods, which seems promising particularly for low-resource languages, is based on sharing acoustic data from multiple resources and representing the target AM by subspace Gaussian mixture models [14]. Last but not least, it should be mentioned that the phonetic transcriptions can be omitted, if the AM is built on graphemes rather than on phonemes. The results published for Russian [15] or Slovak [16] show, however, that the classic phoneme-based HMMs always outperform the grapheme-based ones.

3. Developing AM in Efficient Way

The goal of our work is to develop AMs for various Slavic languages with minimum costs. Yet, we want the performance of these AMs to be as high as possible, as it will allow us to use unsupervised training and adaptation techniques in later stages when more data is available.

We start the AM process building with bootstrapping from a Czech phoneme-based AM and then we utilize two schemes. One is applicable to speech data with orthographic (but not necessarily error-free) annotations, and its main goal is to use the existing LVCSR system to generate phonetic transcriptions, to check these annotations and identifying possible inconsistencies in them. The other approach is based on searching for publically available audio data that contain speech and for which some additional text information (e.g. in form of summaries, captions or quotations) can be found. By matching the text resources with the output of the LVCSR we identify the portions with a high level of agreement and utilize them for iterative retraining of the target AM.

3.1 Generating Phonetic Transcriptions for Imperfect Speech and Text Data

When using a speech database provided by a third party, we should be prepared for the situation that not all audio and text data are perfect. Some major errors, like missing files, missing parts of utterances or their transcriptions, can be discovered early, but smaller errors caused either by speakers or annotators are hard to be detected without an expert in the target languages. If the degree of inconsistence is high (which may be true even for some established databases as shown in Section 4), a straightforward application of the forced alignment technique would not be the best option for generating phonetic transcriptions. In this case, it is important to apply a procedure that is capable of checking the audio and reference text content and identifying potential problems. In our scheme, it is the developed target LVCSR system itself that plays the role of the expert who automatically checks the files, generates the transcriptions, and provides hints to a human supervisor where he or she should intervene.

3.1.1 Basic Scheme

The scheme runs iteratively, with the following steps:

1. Preparation. It consists in preparing the existing LVCSR for running with the lexicon and LM of the target language. Pronunciations in the lexicon are temporarily mapped to the phonetic inventory of the source language.

2. Initialization. The existing AM from the source languages is used. All files in the database are labeled as *NotChecked*.

3. Transcription. All *NotChecked* files are transcribed using the LVCSR system and the current AM.

4. Matching. For each audio file, the recognizer's output is matched to the reference text. To quantify the agreement on the word level, we use the standard Word Error Rate (WER) measure:

$$WER = (N_S + N_D + N_I)/N.100\% \quad (1)$$

where N_S, N_D, N_I are the numbers of substitutions, deletions, insertions and N is the total number of words, respectively. As the recognizer produces also a phonetic transcription, we can match it to the phonetic transcription generated from the reference text using the lexicon or a grapheme-to-phoneme transducer (G2P). By applying the same matching procedure as for the words, we get a similar measure denoted as Phoneme Error Rate (*PER*).

5. Classification. If an utterance yields $WER = 0$, it is (almost) sure that the reference text is correct, the audio

file is uncorrupted and the automatically generated phonetic transcription is appropriate. If $WER > 0$ but $PER = 0$, it means that the text disagreement is caused either by homophones or spelling variants. Each utterance is classified into one of the three classes: *Accepted* (if $WER = 0$ or $PER = 0$), *ToBeChecked* (if $WER < T_W$) and *NotChecked* (otherwise).

6. Manual Check. The *ToBeChecked* utterances are those with little disagreement. It can be 1 to 3 words (depending in the utterance length) when we set threshold value $T_W = 10$ %. Using a simple check program, we can visualize the differences, listen to them and correct either the reference text or the LVCSR output (both the orthographic and the phonetic one). After that, the checked utterance get label *Accepted*.

7. Checkpoint. If there are no new *Accepted* utterances in the current iteration, the procedure is stopped here.

8. Retraining. A new AM is trained using phonetic transcriptions of all *Accepted* utterances.

9. Repeat. The procedure goes back to step 3.

3.1.2 Enhanced Scheme

The above described scheme can be further improved to get faster progress with less human work.

a) The WER values will be reduced if the lexicon and LM used in step 1 are better fitted to the given speech database. It can be done by adding (temporarily) the database specific words to the lexicon and similarly by adding the reference texts to the LM training corpus.

b) If we are not sure about the pronunciation of some words in the target language, or about the proper mapping of target language phoneme set, we can use multiple variant pronunciations and let the recognizer decide which one is more appropriate or statistically more frequent.

c) Although the amount of human work needed in step 6 is significantly smaller compared to full manual check, it still can be reduced if more utterances get the *Accepted* label. This can be achieved by utilizing several different AMs in step 3 of each iteration. These AMs can differ only slightly, e.g. by the number of HMM mixtures or by using global or sliding-window Cepstral Mean Subtraction (CMS) parameterization. It is possible to use also an AM which is trained on the mix of the already *Accepted* utterances (from the target language) with some amount of speech from the source language. It is very likely that each of these different AMs will produce a slightly different set of *Accepted* utterances, and hence their total number in each iteration will be increased. We demonstrate the positive effect of this idea in Section 4.

d) If the size of the training data in the target language is large enough, the transcriptions and the lexicon are remapped back to the original phoneme set. (After this step, however, only the target language AM can be employed.)

3.2 Automatic Speech and Text Data Mining from Web

The main problem of training speech databases is that their size is limited and they are available only for some languages. Though, one can find a lot of audio files containing speech on Internet, e.g. in publicly accessible archives of radio and TV broadcasters, on web pages of some institutions, like parliaments, senates, courts, etc. In some cases, these audio files are accompanied also with texts. The ideal situation occurs when these texts are verbatim transcriptions of speech files. In this trivial case, no special procedure is needed to align them and to make them a part of the training data. In most cases, however, the text differs to some extent from the speech. A high degree of correspondence occurs, e.g. between broadcast speech and attached close captions, which has been often utilized for so called lightly supervised AM training (e.g. in [17]). In other cases, the accompanying texts may be just summaries of what was spoken, news articles containing quotations, or documents that were discussed e.g. during a parliament session, etc. However, even these loosely related texts can serve for collecting data suitable for AM training.

If a source (usually a web page) containing both text and speech is found, one of the four situations illustrated in Fig. 1 can occur: a) the text has nothing in common with the audio file, b) the text and the speech share some common words (usually prepositions, conjunctions, pronouns) that are randomly scattered, c) the text and speech contain coincident phrases (strings of few words), and d) the two sources are related in the way, that some spoken utterances occur as written (not necessarily verbatim) sentences in the text. The last case is a good opportunity for automatic acquisition of new training data. Yet, most of the potential Internet sources are mixes of the four cases, with case d) often being the least frequent one. Anyway, if the source is large and the used data mining method is robust, we can collect a considerable amount of new training material.

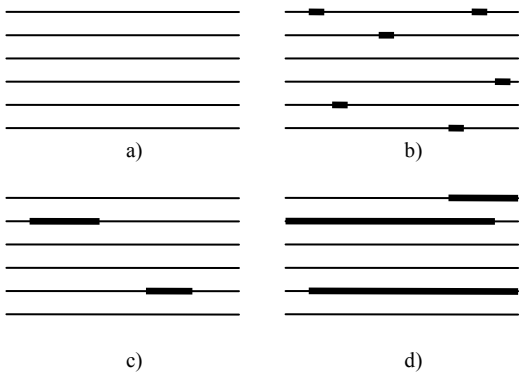


Fig. 1. Web page text and its parts found in audio (bold) a) no correspondence, b) randomly scattered words, c) shorter phrases, d) longer utterances.

LVCSR *raw*	...	skijašice	[noise]	startaju	sutra	prva	vožnja	počinje	u	15			druga	u	18	[noise]	muški	…
LVCSR w_j	…	skijašice		startaju	sutra	prva	vožnja	počinje	u	15			druga	u	18		muški	…
TEXT r_i	…	skijašice		startaju	sutra	prva	vožnja	počinje	u	15	sati	a	druga	u	18			…
LABEL		H	-	H	H	H	H	H	H	H	D	D	H	H	H	-	I	

START STOP

Fig. 2. Example of alignment of a part of LVCSR output (raw and text-only) with a part of reference text (in Croatian). Here, 2 words occurring in text were not found by LVCSR (words "sati" and "a" were not spoken) and recognized word "muški" did not appear in text. Symbols START and STOP denote endpoints of an eligible word sequence. The actual cut points are moved to the nearest occurrence of silence.

3.2.1 Searching for Speech Related to Text

The first step consists in finding a large publicly accessible Internet source that is structured into smaller units, usually web pages, containing audio files and some text. Then, for each page, we search if there are speech and text segments that correspond to each other. The search is based again on matching the text (it will be further referred to as a *reference*) to the output of the currently available LVCSR system. For the alignment of the two strings we have proposed a variant of the Minimum Edit Distance (MED) algorithm, inspired by [18], that prefers local rather than global alignment of word sequences.

The algorithm searches for the optimal alignment of two sequences, the reference comprised of J words r_j and the recognizer output comprised of I words w_i (numbers I and J can differ significantly). This type of tasks is usually solved by the dynamic programming approach, using distance matrix **A** as a space where the solution is searched. The procedure starts with initialization:

$$\begin{aligned} A(i,0) &= P_D \cdot (i-1), 1 \le i \le I \\ A(0,j) &= P_I \cdot (j-1), 1 \le j \le J \end{aligned} \tag{2}$$

and continues with the recursive computation of all other $A(i,j)$ values:

$$\begin{aligned} A(i,j) = \min[&A(i-1,j-1) + d(r_i, w_j) - b_{i-1\,j-1}; \\ &A(i,j-1) + P_I; A(i-1,j) + P_D] \end{aligned} \tag{3}$$

where

$$d(r_i, w_j) = \begin{cases} 0 & \text{if } r_i = w_j \\ P_S & \text{if } r_i \ne w_j \end{cases} \tag{4}$$

and

$$b_{ij} = \begin{cases} 0 & \text{if } r_i \ne w_j \\ b_{i-1\,j-1} + 1 & \text{if } r_i = w_j \end{cases} \tag{5}$$

Constants P_D, P_I and P_S are penalties associated with word deletion, insertion and substitution, respectively. Their optimal values depend on which of the cases shown in Fig. 1 are typical for the source data. We use a small subset of this data to determine them experimentally. Auxiliary values b_{ij} help us to keep a track of non-interrupted hit sequences. In (3), these obtain a small bonus.

When all cells in matrix **A** are computed, the best alignment path is revealed by a standard backtracking procedure from final point (I,J) to starting point (1,1). Each word in reference is assigned one of the labels: Hit (H), Substitution (S), Deletion (D) or Insertion (I).

The next step consists in identifying word sequences where the reference text and the LVCSR output are either same or differing only slightly. The algorithm goes through the word labels and searches for sequences with dominating hits. Formally expressed, we search for a string of words W_1, W_2,... W_N that meets the following constraints:

$$\begin{aligned} &N_{\min} < N < N_{\max} \\ &Label(W_1) = Label(W_2) = Label(W_N) = Hit \\ &N_H > (N_S + N_I + N_D) \end{aligned} \tag{6}$$

The sequence should have minimum length N_{min} and for practical reasons it should not be longer than N_{max} (otherwise it is split). The first, second and last words must be labeled as hits, and the total number of hits N_H in the sequence should be higher than the rest. The last constraint may seem weak but let us note that at this level we search for data that will be later processed with an LVCSR system whose performance will improve in time and some non-hit terms get a chance to be classified correctly.

In the last step, the utterances belonging to the eligible sequences are cut out from the original (often very long) audio files and stored with the corresponding text. The cut points are derived from the time stamps associated to each word (and non-speech event) during the LVCSR procedure. To minimize problems with inaccurate cuts at the beginning and end of the utterance, the actual cut points are moved to the center of the nearest noise event (usually silence or breath). The whole process is illustrated in Fig. 2.

3.2.2 Making Training Database from the Mined Data

After completing the process described above, we get a collection of audio files with reference texts, i.e. data

similar to many standard speech databases. Obviously, we must be aware of the fact that the text transcriptions can contain errors but the same happens also to official databases. What is missing is information about speakers. Yet, there are ways to cope with this problem. In case of broadcast archives, many web pages mention the name of the editor, who often is the main speaker in the audio file. On parliament or senate pages, the speaker name is often explicitly stated. Another alternative consists in utilizing speaker recognition methods to identify different speakers. This speaker clustering is necessary if we want to apply a limit for the amount of the data provided by a single speaker. After this step, the speech database is ready for the process described in Section 3.1.

4. Evaluation on Croatian LVCSR

The methods proposed in the previous section have been successfully applied for the development of LVCSR systems in four languages (Slovak, Russian, Croatian and Polish). In the following text we will focus only on their evaluation on Croatian, as it has been the most challenging language so far, mainly because of very limited resources.

4.1 LVCSR System Applied to Croatian

The evaluation experiments were conducted on the standard LVCSR system originally developed for Czech and recently described e.g. in [19]. Its front-end processes 16 kHz audio data, converts them into 39 MFCC features, applies global or floating CMS, and HLDA. The Czech AM uses triphone HMMs to represents 41 Czech phonemes and 7 types of noise. Its recent version has been trained on 320 hours of speech (of various types). The decoder runs in real-time with vocabularies up to 500K words and a bigram LM smoothed by Kneser-Ney method is used in standard one-pass mode.

When preparing the system for Croatian, we collected from Internet a large corpus (940 MB) of newspaper text. We used it to compile a 255K lexicon and a bigram LM based on 28M different word-pairs. Three Croatian specific phonemes (represented by graphemes 'ć', 'đ' and 'lj') were mapped to the closest Czech counterparts. More details on these basic preparation steps can be found in [20].

4.2 Speech Data for Training and Testing

In this study we used 3 sources of Croatian data, the GlobalPhone set, the COST set and the HRT web resource.

4.2.1 GlobalPhone - HR

This data is part of a large multi-lingual speech corpus collected by the team at the University of Karlsruhe [11]. Recently it includes 20 languages from various parts of the world and its subsets are distributed on commercial base via ELRA [21]. Unlike the other language sets in the GlobalPhone (GP) collection, the Croatian one has some specific features. First, its size is smaller compared to the other sets. It contains 4499 utterances that were recorded by 92 speakers. (Most other language sets contain about 10,000 recordings from 100 speakers). Second, the distribution of the recordings among the speakers is not balanced, as some speakers recorded less than 30 sentences while some others contributed more than 100 ones. Third, the speech is supposed to be read but in many cases the speakers did not read given sentences fluently, they mispronounced words, repeated them, made false starts, or they uttered words different from those in the text form. These mistakes and the fact that most speakers were actually speaking Bosnian (using different words, e.g. 'hiliada' instead of 'tisuća', and slightly different pronunciation) complicates automatic processing of the recordings. Obviously, the database as it is can be used for training the AM applicable for Croatian LVCSR experiments as it was shown in [5]. However, in this case, native human annotators (who are able to discover and fix the errors) are necessary.

4.2.2 COST278 - HR

This is another multi-lingual speech database. It was created within European COST278 project to support international collaboration on broadcast news processing, namely in speaker segmentation and clustering tasks [22]. It includes 5 to 10 complete TV shows in 9 languages (about 3 hours per each), including Croatian. Each show is manually segmented and orthographically transcribed.

4.2.3 HRT Radio Speech Data

When searching for additional speech resources we discovered the web archive of the major public broadcaster in Croatia, HRT. Its regional stations have their own web sites, with pages devoted to short local news. The news is described by text and occasionally also by audio. In most cases, the correspondence between the text and speech resembles situations a), b) or c) illustrated in Fig. 1. However, the amount of available audio (several hundred hours) and text (about 10K files) allows for experimenting with the method proposed in Section 3.2. For this purpose we have chosen data covering the 1/2010 to 7/2012 period.

4.2.4 Test Sets

Set	Speech style and recording year	Size in minutes	#words	OOV [%]
GP	speech produced by amateurs (1998)	59	7386	1.96
COST	read/planned speech by professionals (2003)	35	5052	1.18
HRT	read/planned speech by professionals (2013)	27	4088	1.13

Tab. 1. Description of three Croatian test sets.

For evaluation, we used the following test data: a) utterances of speakers 02, 03, 04, 06, and 07 from the GlobalPhone set, b) 307 speech segments from 2 COST278

TV shows and c) 104 utterances mined from HRT radio station Pula (news broadcasted in January 2013).

4.3 Bootstrapping with Czech AM

In the first series of experiments, we tried to measure, what performance can be achieved with a purely Czech AM. The second question was which type of Czech AM is optimal for bootstrapping a Croatian system. We compared an AM tuned for the best performance in Czech LVCSR with two AMs represented by a lower number of parameters (physical states). The results are in Tab. 2.

	WER [%] for 3 test sets		
AM parameters	**GP**	**COST**	**HRT**
5575 states, discrim. training	32.36	25.12	28.40
4044 states, EM training	32.07	24.78	26.54
2041 states, EM training	28.47	23.65	26.39

Tab. 2. Performance achieved with 3 Czech acoustic models.

The figures show that all AMs, and especially those less fitted to Czech, have an acceptable performance in the initial tests. We concluded that for bootstrapping, the 2041-state model was the optimal choice.

4.4 AM Trained on Croatian GlobalPhone Set

As explained earlier, the Croatian GlobalPhone set is a really challenging speech resource. Its precise phonetic annotation would be a difficult task even for a native speaker or a skilled phonetician. The main problems stem from low acoustic quality of some recording sessions, influent speech interrupted by many restarts, incorrect orthographic transcriptions, inconsistent pronunciation and the use of Bosnian language by more than half speakers. There are also occasional background voices or audible prompts from the recording supervisors. Hence, the application of the transcription method described in Section 3.1 promises to save a lot of tedious manual work.

Before launching the proposed iterative procedure we slightly adapted the general-purpose lexicon and the LM to better fit the GlobalPhone utterances. This step was necessary, as the database was recorded in 1998 and most utterances deal with war and post-war events of that period. About 200 most frequent OOV words were added to the vocabulary and all sentences (except those used for testing) were included in the LM training corpus.

The transcription process run according to the enhanced scheme described in Section 3.1. In each iteration loop, we used several available AMs. To illustrate their effect, let us compare performance of two of them: one based entirely on the already transcribed Croatian data, the other trained on the mix of the same data and 10 hours of randomly chosen Czech training sentences. In Tab. 3, we can observe that the additional Czech data helped to improve the AM and to reduce WER values, especially in initial stages. Another advantage of the mixed model is that its Czech part supplies training data for (so far) rarely seen phoneme context and, in particular, for noise models. Not only these two AMs but also their variants differing in numbers of mixtures (32 or 16) or in the application of the CMS normalization (global or floating) were utilized in each iteration.

	WER [%]									
Hours	1	2	3	4	5	6	7	8	9	10
HR	29.2	24.9	23.7	24.4	22.1	21.8	21.4	20.7	20.3	20.0
HR+CZ	25.3	22.8	21.5	21.7	21.0	20.5	20.7	20.5	19.8	19.7

Tab. 3. WER obtained on GP test set for AMs trained on increasing amounts of Croatian speech (HR) and with 10 hours of Czech (HR+CZ).

The amount of transcribed data after each iteration is shown in Fig. 3. We can see that, e.g. after the first iteration (in which bootstrapped Czech AMs were used), we got 1.7 hours of phonetically annotated data. From this amount, 1.3 hours were obtained automatically, 0.4 hours required small manual corrections related to 1 or 2 words (either in the reference text or in the LVCSR output). We can also notice that the largest gain occurred during the first 4 iterations. The process was stopped after the 12th iteration, when 11.5 hours were transcribed. The remaining amount (1.7 hours) from all the 13.2 hours allocated for training was not used, as these data had either bad acoustic quality or they were hard to be corrected by a non-expert. The whole procedure consumed mainly computer time, while the total amount of required manual work took just a small portion of it.

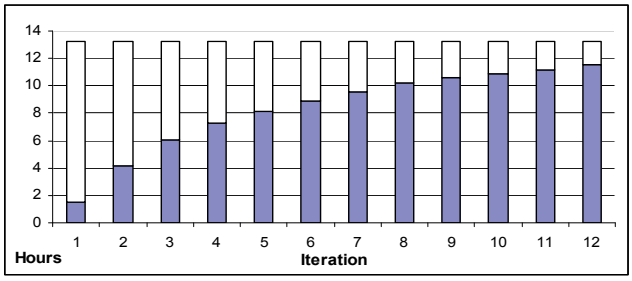


Fig. 3. Amount (in hours) of transcribed GlobalPhone data during the iterative procedure described in Section 3.1.

In each iteration step there were approx. 100-200 utterances that passed the threshold $T_W = 10\,\%$ and that were eligible for human check. We designed a simple tool that highlights the difference between the reference text and the LVCSR output. The tool can play either the whole utterance or the selected part. In most cases, the human supervisor just needs to decide which word is correct (either the reference or the recognized one) and make the adjustment by one click. No prior knowledge of the target language is needed for this type of action. If the annotator is not sure about the correct word (because it is unknown to him/her, or its pronunciation is unclear or incomplete, or it is masked by noise, etc) he/she can skip the utterance and remove it from further processing. One of the biggest benefits of this scheme is that the human work is focused only on those utterances that require minimum effort. The time spent by correcting the Croatian recordings labeled as *ToBeChecked* was about 2-3 hours in each iteration.

The effect of the AM trained on these 11.5 hours was evaluated on the 3 test sets. The results presented in Tab. 4 can be directly compared with those in Tab. 2. We can see that the improvement in performance is significant.

	WER [%] for 3 test sets		
AM	**GP**	**COST**	**HRT**
Trained on 11.5 hours of GP data	19.93	16.29	18.13

Tab. 4. WER values obtained with AM trained on GlobalPhone data.

4.5 AM Trained on HRT Radio Archive Data

The goal of this experiment was to verify how good can be an AM that is trained entirely on data automatically collected from web. As explained in Section 4.2.3, our source was HRT radio archive. We found 11,851 web pages that contained both audio files and text. On a small subset of these data (200 pages randomly chosen) we analyzed the correspondence between the audio and text content. Unfortunately, the most favorable case (that depicted in Fig. 1d) was very rare. In most cases, the alignment between the text and speech revealed no common sequences. During a parameter optimization process, we set the following constants: $P_S = 15$, $P_D = 10$, $P_I = 3$, $N_{min} = 10$ and $N_{max} = 25$. Using the bootstrapped Czech AM we run the method described in Section 3.2. It found 3,694 segments that met the constraints defined by (6). Their total duration was 8.8 hours. After that, this data passed through the same iterative procedure as applied to the GlobalPhone data. In this case, only 5 iterations were necessary to transcribe (mostly automatically) the complete set and to train the final AM. The smaller number of iterations can be explained by the fact that the selected segments contained mainly clean speech produced by professionals in studio. Moreover, many manual interventions dealt with the reference text rather than the recognized one.

The results achieved with this AM are listed in Tab. 5. When comparing them to those in Tab. 4, we can see that with one exception (the GP test set), the performance is better, in spite of a smaller amount of the training data. Let us also remind that this data are cost-free. Obviously, the process of the audio data mining could be repeated with the new Croatian AM and it is expected that more data would be acquired.

	WER [%] for 3 test sets		
AM	**GP**	**COST**	**HRT**
Trained on 8.8 hours of HRT data	19.98	14.89	15.14

Tab. 5. WER values obtained with AM trained on HRT data.

4.6 AM Trained on All Available Data

In the last experiment, we made a natural step and put all the available training data together: GlobalPhone (11.5 hours), HRT (8.8 hours) and 1.3 hours acquired through the same transcription scheme from the remaining part (3 TV shows) of the COST database. We trained the final AM on these 21.6 hours of Croatian data coming from three different sources and three time periods (1998, 2003, 2010-2012). The results are summarized in Tab. 6. We can notice a consistent improvement for all the three test sets.

	WER [%] for 3 test sets		
AM (# physical HMM states)	**GP**	**COST**	**HRT**
Trained on 21.6 hours of HRT, GP and COST data (1541states)	17.55	14.12	14.28

Tab. 6. WER values obtained with AM trained on all available data.

5. Discussion and Conclusions

We have proposed and evaluated two schemes that can save a significant portion of human work in developing acoustic models for languages that are related to one with an existing AM. Both schemes utilize a LVCSR system as a tool that performs the two functions: The first is to check the validity of orthographic transcriptions that are provided either explicitly, e.g. as a part of a speech database, or that can be acquired from public sources like Internet. The second function is to generate phonetic transcriptions by using a lexicon (or a G2P transducer), choosing between alternative pronunciations, and identifying and labeling non-speech sounds.

We have also shown that an acoustic model for a new language can be trained without a dedicated, commercially distributed speech database. The data we acquired automatically from publicly available Internet sources enabled us to train an AM whose performance is better than that made of the Croatian part of the GlobalPhone database.

Both schemes have been already used in practice: for Croatian - as documented in this paper - and also for Slovak, Russian and Polish. (Let us note that the quality of the Russian and Polish GlobalPhone subsets was significantly better compared to the Croatian one.) The availability of the AMs for the other Slavic languages allows us to further enhance the proposed methods, for example by utilizing multiple and multi-lingual acoustic models within the bootstrapping phase. To examine the idea, we have run a simple experiment, in which five AMs, each developed for one language, were tested on the three Croatian sets. From the results presented in Tab. 7, we can observe that the Slovak AM would be even better in the bootstrapping phase than the Czech one was.

	WER [%] for 3 test sets		
AM (# physical HMM states)	**GP**	**COST**	**HRT**
Czech (2041 states)	28.47	23.65	26.39
Slovak (3764 states)	26.09	19.58	22.36
Polish (2035 states)	32.37	27.04	26.22
Russian (3382 states)	33.54	27.85	30.35
Croatian (1541 states)	17.55	14.12	14.28

Tab. 7. WER values obtained with AMs representing 5 languages.

The AMs developed for the four Slavic languages represent a good starting point for demonstrating the potential of an LVCSR in tasks like broadcast news tran-

scription. In each of the four languages, we got close to the 15-percent-WER level, at least for read speech. This level allows for running a system that can monitor broadcast news programs and save the data for further AM improvements, via lightly supervised or even unsupervised techniques, which is our main research direction in this field, recently.

Acknowledgements

This work was supported by the Czech Science Foundation (project no. P103/11/P499) and by the Technology Agency of the Czech Republic (project no. TA01011204).

References

[1] GAUVAIN, J-L., LAMEL, L., ADDA G. The LIMSI broadcast news transcription system. *Speech Communication*, 2002, vol. 37, no. 1-2, p.89-108.

[2] LAMEL, L., MESSAOUDI, A., GAUVAIN, J-L. Automatic speech-to-text transcription in Arabic. *ACM Transactions on Asian Language Information Processing*, 2009, vol. 8, no. 4, p. 1-17.

[3] LAMEL, L., VIERU, B. Development of a speech-to-text transcription system for Finnish. In *Proc. of SLTU 2010*. Penang (Malaysia), 2010, p. 62–67.

[4] ADDA-DECKER, M., LAMEL, L., ADDA, G. A first LVCSR system for Luxembourgish, an under-resourced European language. In *Proc. of LTC'11 Workshop*. Poznan (Poland), 2011, p. 47-50.

[5] VU, N. T., SCHLIPPE, T., KRAUS, F., SCHULTZ, T. Rapid bootstrapping of five eastern European languages using the rapid language adaptation toolkit. In *Proc. of Interspeech 2010*. Makuhari (Japan), p. 865–868.

[6] PROCHAZKA, V., POLLAK, P., ZDANSKY, J., NOUZA, J. Performance of Czech speech recognition with language models created from public resources. *Radioengineering*, 2011, vol. 20, no. 4, p. 1002-1008.

[7] SCHULTZ, T., BLACK, A. Rapid language adaptation tools and technologies for multilingual speech processing. In *Proc. of ICASSP 2008*. Las Vegas (USA).

[8] NOUZA, J., ZDANSKY, J., CERVA, P., KOLORENC, J. Continual on-line monitoring of Czech spoken broadcast programs. In *Proc. of Interspeech 2006*. Pittsburgh (USA), p. 1650-1653.

[9] NOUZA, J., et al. Voice technology to enable sophisticated access to historical audio archive of the Czech radio. *Multimedia for Cultural Heritage*. Springer Berlin Heidelberg, 2012, CCIS vol. 247, p.27-38.

[10] NOUZA, J., SILOVSKY, J., ZDANSKY, J., CERVA, P., KROUL, M., CHALOUPKA, J. Czech-to-Slovak adapted broadcast news transcription system. In *Proc. of Interspeech 2008*. Brisbane (Australia), p. 2683-2686.

[11] SCHULTZ, T. GlobalPhone: A multilingual speech and text database developed at Karlsruhe University. In *Proc. of ICSLP 2002*. Denver (USA), 2002, p. 345–348.

[12] SCHULTZ, T., WAIBEL, A. Language-independent and language-adaptive acoustic modelling for speech recognition. *Speech Communication*, 2001, vol. 35, no 1–2, p. 31–51.

[13] LÖÖF, J., GOLLAN, C., NEY, H. Cross-language bootstrapping for unsupervised acoustic model training: Rapid development of a Polish speech recognition system. In *Proc. of Interspeech 2009*, Brighton (UK), p. 88-91.

[14] BURGET, L., et al. Multilingual acoustic modeling for speech recognition based on subspace Gaussian mixture models. In *Proc. of ICASSP'10.* Dallas (USA).

[15] STÜKER, S., SCHULTZ, T. A Grapheme based speech recognition system for Russian. In *Proc of Specom 2004*. St. Petersburg (Russia), September 2004.

[16] MIRILOVIC, M., JUHAR, J., CIZMAR, A. Comparison of grapheme and phoneme based acoustic modeling in LVCSR task in Slovak. *Multimodal Signal: Cognitive and Algorithmic Issues*. Springer, LNAI, vol. 5398, 2009, p. 242–247.

[17] LAMEL, L., GAUVAIN, J-L., ADDA, G. Lightly supervised and unsupervised acoustic model training. *Computer, Speech & Language*. 2002, vol. 16, no. 1, p. 115–129.

[18] HIRSCHBERG, D. S. Algorithms for the longest common subsequence problem. *J. of the ACM*, 1977, vol. 24, no.4, p. 664–675.

[19] NOUZA J., et al. Making Czech historical radio archive accessible and searchable for wide public. *Journal of Multimedia,* 2012*, v*ol. 7, no. 2, p. 159-16.

[20] NOUZA, J., CERVA, P., ZDANSKY, J., KUCHAROVA, M. A study on adapting Czech automatic speech recognition system to Croatian language. In *Proc. of Elmar 2012*. Zadar (Croatia), 2012, p. 227-230.

[21] ELRA catalogue, GlobalPhone - HR, ref. number ELRA-S0195.

[22] ZIBERT, J., et al. The COST278 broadcast news segmentation and speaker clustering dvaluation. Overview, methodology, systems, results. In *Proc. of Interspeech 2005*. Lisbon (Portugal), 2005, p. 628-631.

[23] HRT archive available at http://www.hrt.hr/

About Authors ...

Jan NOUZA (*1957) received his M.Sc. and Ph.D. degrees at the Czech Technical University (Faculty of Electrical Engineering) in Prague in 1981 and 1986, respectively. Since 1987 he has been teaching and doing research at the Technical University in Liberec. In 1999 he became a full professor. His research focuses mainly on speech recognition and voice technology applications, such as voice-to-text conversion, dictation, broadcast speech processing and design of voice-controlled tools for handicapped persons. He is the head of SpeechLab group at the Institute of Information Technology and Electronics.

Petr ČERVA (*1980) received the Master degree and the Ph.D. degree from the Technical University of Liberec (TUL), in 2004 and 2007, respectively. He is currently an assistant professor at the Inst. of Information Technology and Electronics at TUL. His research interests are speaker adaptation and speech recognition.

Michaela KUCHAŘOVÁ (*1987) received her MSc degree in Information Technology from TUL in 2011. She joined SpeechLab as a PhD student and recently she works mainly on linguistic topics, with a special interest in multilingual issues.

Interspeech 2018
2-6 September 2018, Hyderabad

Using Deep Neural Networks for Identification of Slavic Languages from Acoustic Signal

Lukas Mateju, Petr Cerva, Jindrich Zdansky, Radek Safarik

Faculty of Mechatronics, Informatics and Interdisciplinary Studies,
Technical University of Liberec, Studentska 2, 461 17 Liberec, Czech Republic
{lukas.mateju, petr.cerva, jindrich.zdansky, radek.safarik}@tul.cz

Abstract

This paper investigates the use of deep neural networks (DNNs) for the task of spoken language identification. Various feed-forward fully connected, convolutional and recurrent DNN architectures are adopted and compared against a baseline i-vector based system. Moreover, DNNs are also utilized for extraction of bottleneck features from the input signal. The dataset used for experimental evaluation contains utterances belonging to languages that are all related to each other and sometimes hard to distinguish even for human listeners: it is compiled from recordings of the 11 most widespread Slavic languages. We also released this Slavic dataset to the general public, because a similar collection is not publicly available through any other source. The best results were yielded by a bidirectional recurrent DNN with gated recurrent units that was fed by bottleneck features. In this case, the baseline ER was reduced from 4.2% to 1.2% and C_{avg} from 2.3% to 0.6%.

Index Terms: language identification, Slavic languages, deep neural networks, convolutional neural networks, recurrent neural networks

1. Introduction

Spoken language identification (LID) is the task of correctly determining the language spoken in a speech utterance. In recent years, many scientific efforts have been dedicated to this task, and nowadays, LID modules form an integral part of many speech processing applications including, e.g., systems for multilingual speech recognition or spoken language translation. LID systems are also used for spoken document retrieval, emergency call-routing or in dialog systems. Although the accuracy of all these systems is constantly improving, it is still not perfect. For example, one of the significant bottlenecks of LID systems is to distinguish between closely related languages.

Most of the state-of-the-art LID systems utilize various advanced acoustic modeling techniques.

One of the most popular techniques relies on the total variability factor analysis, and it is known as an i-vector framework [1, 2]. I-vector is a fixed length representation of an utterance, and it jointly contains information about the speaker, language, etc. (e.g., LDA might be applied to obtain discriminative features). To extract i-vector features, hand-crafted shifted delta cepstral features (SDC) derived from mel-frequency cepstral coefficients (MFCCs) [3] and phone log-likelihood ratios (PLLRs) [4] are most commonly used as inputs. The i-vector extraction is usually followed by a classification stage, where multiclass logistic regression, cosine scoring or Gaussian models are utilized. The major drawback of the i-vector approach is the decreasing performance on shorter test utterances [5].

Over the past few years, deep neural networks have had an upsurge in popularity in LID systems thanks to their outstanding performance in many other speech processing applications (e.g., speech recognition [6]). Both direct and indirect approaches exist for utilizing deep learning for LID.

In the former case, so-called bottleneck features (BTNs) are widely used in many systems [7, 8, 9] due to their superior performance. Usually, these features are extracted from a DNN trained to discriminate individual physical states of a tied-state triphone model at first, and then used as inputs to an i-vector based system [10, 11].

In the latter case, various end-to-end systems based on different DNN architectures are trained to identify the language in the input utterance. In 2014, feed-forward DNN yielded excellent results on short utterances (less than 3 seconds) [5]. Since then, other more advanced architectures, such as attention based DNNs [12], convolutional neural networks (CNNs) [13, 14, 15], time delay neural networks (TDNNs) [16, 17] or sequence summarizing neural networks (SSNNs) [18] have also been successfully used. The most recent direct approaches take advantage of recurrent neural networks (RNNs) and their context modeling ability. Gated recurrent unit (GRU) RNNs [19], long short-term memory (LSTM) RNNs [20, 21, 22, 23, 24] and bidirectional LSTM RNNs [25, 26] all yield the state-of-the-art performance.

In this paper, various state-of-the-art LID methods are investigated. We adopt feed-forward DNNs at first, then CNNs, and finally also unidirectional as well as bidirectional RNNs with both previously mentioned types of units. We also combine these direct methods with the indirect approach: we feed the networks with bottleneck features. To the best of our knowledge, results of some of these approaches and their comparison on one dataset have not yet been published for LID.

The experimental evaluation is performed on a dataset consisting of the 11 most widespread Slavic languages. These were selected for two main reasons.

The first is that most of these languages are related to each other which makes our dataset more challenging. This is especially true for those pairs of languages that belong to the same language branch. For example, it is difficult to distinguish between Croatian and Serbian (South Slavic branch), even for native speakers.

Secondly, only results obtained for several (pairs) of Slavic languages have been published so far (e.g., [27]). For example, Polish and Russian formed one cluster of related languages within the last Language Recognition Evaluation (LRE) challenge in 2017 [28]. On the contrary, a detailed analysis for all evaluated Slavic languages using a confusion matrix is presented in this work.

Finally, note that our dataset of Slavic languages is available for download to the general public[1].

[1] https://owncloud.cesnet.cz/index.php/s/gXHKFs9UDEqe34G

10.21437/Interspeech.2018-1165

2. Dataset of Slavic languages

Slavic languages are spoken by approximately 320 million people throughout Eurasia mostly in Central, Eastern, and Southern Europe. There are at least ten languages with over a million native speakers (e.g., Russian (∼150 million speakers), Polish (∼55 million speakers), Czech (∼11 million speakers), etc.).

Slavic languages can be divided on the basis of geographical and genealogical principle into three main branches: East, South and West Slavic languages. Most of the Slavic languages belonging to the same branch are somehow close to each other. However, while some languages may have similar phonetics (e.g., Croatian and Serbian are practically identical in their phonetics), some languages may have different phonetics (e.g., Polish or Bulgarian are phonetically somewhat more similar to East Slavic languages than to languages in their branch). Every Slavic language has its unique phonetic inventory which distinguishes it from other languages (except for the previously mentioned Croatian and Serbian). It can help with language identification. Moreover, rich morphology, a high degree of inflection, and more or less free word order result in a large linguistic complexity of all these languages.

Due to the lack of an extensive audio dataset for all Slavic languages, we had to create a new one. It is compiled from recordings belonging to the 11 most widespread Slavic languages so that it covers all three branches:

- East Slavic languages - Belarusian, Russian, Ukrainian,
- South Slavic languages - Bulgarian, Croatian, Macedonian, Serbian, Slovene,
- West Slavic languages - Czech, Polish, Slovak.

The source of data for individual languages varies. A majority of the data originates in TV and radio broadcasts, and it was retrieved as described in detail in [29]. It formerly served for acoustic model training for speech recognition, and thus it contains mostly clean speech. The rest of the dataset is formed by microphone recordings.

The data for each language is compiled from recordings belonging to multiple speakers (with both genders represented). It is divided into two non-overlapping subsets: 20 hours of recordings are available for training and 500 utterances for evaluation. The focus is on short recordings (similar to [5]), the maximal duration of an evaluation recording is 5 seconds.

3. Evaluation metrics

Within the scope of this paper, two different performance metrics (namely error rate (ER) and C_{avg}) were utilized to evaluate the performance of LID approaches.

The first metric, error rate, is defined as:

$$ER[\%] = \frac{M_{\mathrm{utt}}}{N_{\mathrm{utt}}} * 100 \ , \qquad (1)$$

where M_{utt} is the number of misclassified speech utterances, and N_{utt} is the total number of evaluated speech utterances.

The second metric is the official metric of the 2015 NIST Language Recognition Evaluation, C_{avg}. Detailed information about this closed set multi-language cost function and its definition can be found in the 2015 LRE Plan [30].

4. Investigated approaches and results

4.1. Acoustic features used

Three different types of 39-dimensional feature vectors were extracted within all of the following experiments: MFCCs (13-dimensional with Δ and $\Delta\Delta$ coefficients), filter bank coefficients (FBCs) and bottleneck features. Both MFCCs and FBCs were computed using 25 ms frames of the signal with frame shifts of 10 ms. As suggested, e.g., in [7, 8, 9], bottleneck features were extracted from DNN trained to discriminate physical states (senones) of a Czech tied-state triphone acoustic model. This DNN was trained on 270 hours of speech recordings belonging to the Czech language and using hyper-parameters as follows: 5 hidden layers with the third one being the bottleneck layer, 1024 neurons per hidden layer (39 for the bottleneck layer), ReLU activation function (sigmoid for the bottleneck layer), a learning rate of 0.08 and 50 epochs of training.

The input for DNNs consists of 11 consecutive FBC vectors, 5 preceding and five following the current frame. Normalization of these vectors was performed within a 1-second long window.

4.2. Baseline i-vector approach

To set a baseline performance, an i-vector system was trained using a full covariance GMM-UBM based system and logistic regression model. Within this training, MFCCs filtered by voice activity detection were employed, and the final extracted i-vectors were 600-dimensional. Note that this baseline approach follows the lre07 recipe as present in Kaldi ASR[2].

The results yielded by this system are presented in the first row of Table 1. They provide a decent baseline (i.e., ER of 4.2% and C_{avg} of 2.3%) for further experimental work.

4.3. Feed-forward fully connected DNN architecture

The first adopted deep learning architecture was a feed-forward fully connected DNN. Its output was formed by a softmax layer with 11 neurons (this value corresponds to the number of languages). This DNN was trained using Torch framework[3] to directly distinguish between languages (i.e., direct method). The hyper-parameters used for training were similar to those for extraction of the bottleneck features (i.e., 5 hidden layers, 1024 neurons per hidden layer, ReLU activation function, a learning rate of 0.08 and 20 training epochs).

During the classification phase, a probability vector was obtained for each frame of given utterance (i.e., by doing a forward pass). These vectors were then averaged, and the language with maximum average probability was selected as an output.

The obtained results for all three types of considered feature vectors are summarized in Table 1. They show that MFCCs slightly outperformed FBCs, but the difference was rather small. It is also evident that the direct approaches (using both MFCCs and FBCs) did not exceed the baseline i-vector based system. On the contrary, the baseline system was outperformed significantly by bottleneck features (see the fourth row of Table 1). The improvement was over 2 % in ER (from 4.2% to 2.0%) and over 1% in C_{avg} (from 2.3% to 1.1%).

Note that we also performed several experiments (not presented in this paper) with the size of the bottleneck layer, but no further reduction in error rate was obtained.

[2] http://kaldi-asr.org/

[3] http://torch.ch/

Table 1: *Results of feed-forward fully connected DNN for different types of features in comparison to baseline i-vector system.*

approach	ER [%]	C_{avg} [%]
LR + i-vectors	4.2	2.3
DNN + MFCCs	5.7	3.1
DNN + FBCs	5.9	3.3
DNN + BTNs	2.0	1.1

4.3.1. The influence of context window size

The next experiment was focused on the importance of the size of input feature context window. The reason is that additional context information may be beneficial for reduction of the error rate of the system. On the contrary, broader context slows down the training and evaluation phases. Several DNNs with a variety of context window sizes from 5-1-5 (i.e., 0.1 seconds long window) up to 50-1-50 (i.e., 1 second) were trained using bottleneck features and evaluated.

The reached results are summarized in Table 2. They show that our initial context window size was too short and degraded the performance. The ideal context window size seems to be longer and around 15-1-15 (i.e., 0.3 seconds). In this case, ER decreased from 2.0% to 1.2% and C_{avg} from 1.1% to 0.7%.

Table 2: *Results for different context window size in the system with bottleneck features and feed-forward fully connected DNN.*

context window size	ER [%]	C_{avg} [%]
5-1-5	2.0	1.1
10-1-10	1.3	0.7
15-1-15	1.2	0.7
20-1-20	1.2	0.7
25-1-25	1.5	0.8
50-1-50	6.6	3.6

4.4. CNN architectures

The next type of DNN networks we focused on were convolutional networks. In contrast to [13], we also tried to utilize the bottleneck features.

The employed CNNs were composed of two convolutional layers and three fully connected layers (each with 1024 neurons). The inputs consisted of 31 feature maps (i.e., context window size of 15-1-15), each 39×1 in size. Our experiments were performed with FBCs and BTN features. The first convolutional layer was comprised of 105 feature maps 39×1 in size followed with a 3:1 max-pooling layer. The second consisted of 157 feature maps 13×1 in size. The rest of the hyper-parameters was set as stated in Sect. 4.3, and the CNNs were also trained using Torch framework.

To explore deeper configurations of CNNs, an additional max-pooling and third convolutional layer (209 feature maps 13×1 in size) were added, and the CNNs were trained.

The achieved results are depicted in Table 3. As expected, the BTN features outperformed FBCs by a large margin (by almost 4%). The more interesting fact is that the difference in performance of DNNs and CNNs was practically negligible. There was no gain in using more complex architecture. The deeper configuration of CNN only worsened the results.

Table 3: *Results of architectures based on CNNs.*

approach	conv. layers	ER [%]	C_{avg} [%]
CNN + FBCs	2	4.9	2.7
CNN + BTNs	2	1.3	0.7
CNN + FBCs	3	6.4	3.5
CNN + BTNs	3	1.7	0.9

4.5. RNN architectures

The last DNN architecture we explored was the recurrent neural network. At first, we focused on long short-term memory RNN architecture (e.g., [20, 31]), but unlike these cited papers, we also investigated the use of bottleneck coefficients. After that, we examined the gated recurrent unit RNNs [19]. Finally, we also explored the possibilities of bidirectional RNNs. We studied slightly different configurations of bi-LSTM RNNs as in [26].

The RNNs were comprised of two recurrent layers (each with 1024 neurons) and two fully connected layers (1024 neurons per layer). The inputs were once again FBCs, and BTN features with a context window size of 15-1-15. The rest of the hyper-parameters remained the same as in Sect. 4.3. The RNNs were trained using the ADAM optimizer in PyTorch[4]. Note that for unidirectional models, the final language for each utterance was obtained by averaging only last 10% of frame probabilities, as suggested in [31], to exploit the learning capabilities of RNNs.

The results are summarized in Table 4. First, as in previous experiments, BTN features outperformed the FBCs. However, the difference in performance was distinctly smaller than that of CNN & DNN architectures. Recurrent neural networks can thus extract more information about the target language from standard acoustic features. Secondly, GRU RNNs exceeded the LSTM RNNs in performance (e.g., 1.4% vs. 1.2% ER). Next, the bidirectional RNNs performance was mixed. Although the bi-LSTM RNNs performed slightly worse than the unidirectional equivalent, the bi-GRU RNNs outperformed its counterpart. However, the gain in performance was rather small (less than 0.1% in ER and 0.1% in C_{avg}). Furthermore, the difference in results between feed-forward fully connected DNN and bi-GRU RNN was rather low as well (0.1% in C_{avg}). The bi-GRU RNN yielded slightly better results but at the cost of more complex architecture to train and evaluate. Finally, the performance of the bidirectional GRU RNN was the best throughout this paper.

Table 4: *Performance of systems based on RNNs.*

approach	ER [%]	C_{avg} [%]
LSTM + FBCs	3.0	1.7
LSTM + BTNs	1.4	0.7
GRU + FBCs	2.5	1.4
GRU + BTNs	1.2	0.7
bi-LSTM + FBCs	3.0	1.6
bi-LSTM + BTNs	1.5	0.8
bi-GRU + FBCs	2.4	1.3
bi-GRU + BTNs	1.2	0.6

[4] http://pytorch.org/

	CZ	SK	PL	RU	SI	UA	RS	MK	HR	BY	BG
CZ	472	10	4	3	1	0	0	0	9	1	0
SK	5	483	5	0	0	0	3	1	0	3	0
PL	9	5	478	5	0	0	0	0	1	1	1
RU	5	3	9	470	1	2	0	0	8	1	1
SI	0	1	0	0	479	4	11	0	2	1	2
UA	1	0	1	1	0	481	0	0	3	9	4
RS	0	3	0	0	10	2	477	1	5	0	2
MK	0	1	0	0	0	2	4	489	1	0	3
HR	8	1	2	5	1	0	6	0	476	0	1
BY	0	1	1	0	4	15	0	2	2	471	4
BG	0	0	0	0	3	0	2	2	0	0	493

	CZ	SK	PL	RU	SI	UA	RS	MK	HR	BY	BG
CZ	497	3	0	0	0	0	0	0	0	0	0
SK	3	489	4	0	1	0	0	0	2	1	0
PL	1	3	493	1	0	0	0	0	1	0	1
RU	3	1	5	490	0	0	0	0	1	0	0
SI	0	0	0	0	495	1	3	1	0	0	0
UA	0	0	0	0	0	495	0	1	0	4	0
RS	0	2	0	0	5	1	484	0	8	0	0
MK	0	1	0	0	0	1	0	493	1	0	4
HR	2	0	0	3	0	0	4	0	491	0	0
BY	0	0	0	0	0	11	1	0	0	487	1
BG	0	1	0	0	0	0	0	1	0	1	497

Figure 1: *Comparison of confusion matrices produced by baseline i-vector system (left) and the best bi-GRU RNN system (right). (CZ - Czech; SK - Slovak; PL - Polish; RU - Russian; SI - Slovene; UA - Ukrainian; RS - Serbian; MK - Macedonian; HR - Croatian; BY - Belarusian; BG - Bulgarian)*

4.6. Error analysis and confusion matrices

More detailed results obtained in the form of confusion matrices are depicted in Figure 1. They show that most of the errors are confusions between related languages. These are mostly caused by a common phonetic inventory but also because the languages have some common words in vocabulary and similar phonotactics, as a wider context is used for identification.

For example, the highest value of errors is between Belarusian and Ukrainian. These languages are more similar to each other than to Russian as they phonetically differ only in a few phonemes and they have similar vocabularies. A comparable case is between Croatian, Serbian and Slovene since the first two have the same phonetic inventory and Slovene differs from both of them only in few phonemes. Vocabularies of these languages are also very similar. Also all west Slavic languages are confused with each other. However, their phonetic inventories are not so close, and the source of confusion may be similar vocabularies and phonotactics.

On the other hand, in some other cases, the confusions are harder to explain and may lay in the nature of recordings (e.g., the source of recordings, acoustic conditions, speaker characteristics) rather than in closeness of the languages. A good example is errors between Russian and Polish as Russian is much closer to other East Slavic languages. Note that some of these confusions, e.g., Croatian with Czech and Russian occurring for the baseline i-vector system, diminished with the use of bi-GRU RNN system.

5. Conclusions

From all the above-stated results, the following conclusions can be drawn: 1) Bottleneck features are beneficial for all investigated DNN architectures, namely for fully connected networks, and yielded the lowest error rates in all scenarios. 2) Without the use of these features, the baseline i-vector based system was able to outperform systems with fully connected DNNs as well as CNNs. 3) The best results were obtained by using bidirectional RNNs with GRU units; however, the relative improvement over the same, but unidirectional system, was small. 4) The evaluation set consisted of recordings no longer than 5 seconds so that the resulting configuration may be utilized even for short recordings.

The more detailed analysis of results in the form of confusion matrices further showed that: 1) According to assumptions, the worst results were in most cases reached for pairs of languages that are related to each other and belong to the same branch of Slavic languages (i.e., they are also difficult to distinguish for humans). 2) The most challenging pair for identification is Belarusian and Ukrainian (East branch). 3) Other more difficult groups of languages to distinguish are Czech, Slovak and Polish (West branch) and Serbian, Croatian and Slovene (from South branch). 4) The resulting RNN-based system was able to reduce mistakes for pairs of languages with low as well as high baseline error rates (i.e., throughout the whole confusion matrix).

6. Acknowledgements

This work was supported by the Technology Agency of the Czech Republic (Project No. TH03010018), and by the Student Grant Scheme 2018 of the Technical University in Liberec.

7. References

[1] N. Dehak, P. A. Torres-Carrasquillo, D. A. Reynolds, and R. Dehak, "Language recognition via i-vectors and dimensionality reduction," in *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011*. ISCA, 2011, pp. 857–860.

[2] D. M. González, O. Plchot, L. Burget, O. Glembek, and P. Matejka, "Language recognition in ivectors space," in *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011*. ISCA, 2011, pp. 861–864.

[3] E. Singer, P. A. Torres-Carrasquillo, D. A. Reynolds, A. McCree, F. Richardson, N. Dehak, and D. E. Sturim, "The MITLL NIST LRE 2011 language recognition system," in *Odyssey 2012: The Speaker and Language Recognition Workshop, Singapore, June 25-28, 2012*. ISCA, 2012, pp. 209–215.

[4] L. F. D'Haro, R. de Córdoba, C. S. Palacios, and J. D. Echeverry, "Extended phone log-likelihood ratio features and acoustic-based i-vectors for language recognition," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*

2014, Florence, Italy, May 4-9, 2014. IEEE, 2014, pp. 5342–5346.

[5] I. Lopez-Moreno, J. Gonzalez-Dominguez, O. Plchot, D. Martinez, J. Gonzalez-Rodriguez, and P. J. Moreno, "Automatic language identification using deep neural networks," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014*. IEEE, 2014, pp. 5337–5341.

[6] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. Audio, Speech & Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[7] M. McLaren, L. Ferrer, and A. Lawson, "Exploring the role of phonetic bottleneck features for speaker and language recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*. IEEE, 2016, pp. 5575–5579.

[8] L. Ferrer, Y. Lei, M. McLaren, and N. Scheffer, "Study of senone-based deep neural network approaches for spoken language recognition," *IEEE/ACM Trans. Audio, Speech & Language Processing*, vol. 24, no. 1, pp. 105–116, 2016.

[9] F. Richardson, D. A. Reynolds, and N. Dehak, "A unified deep neural network for speaker and language recognition," in *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*. ISCA, 2015, pp. 1146–1150.

[10] Y. Song, X. Hong, B. Jiang, R. Cui, I. V. McLoughlin, and L. Dai, "Deep bottleneck network based i-vector representation for language identification," in *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*. ISCA, 2015, pp. 398–402.

[11] R. Fér, P. Matejka, F. Grézl, O. Plchot, and J. Cernocký, "Multilingual bottleneck features for language recognition," in *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*. ISCA, 2015, pp. 389–393.

[12] M. K. V., S. Achanta, L. H. R., S. V. Gangashetty, and A. K. Vuppala, "An investigation of deep neural network architectures for language recognition in indian languages," in *INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*. ISCA, 2016, pp. 2930–2933.

[13] A. Lozano-Diez, R. Zazo-Candil, J. Gonzalez-Dominguez, D. T. Toledano, and J. González-Rodríguez, "An end-to-end approach to language identification in short utterances using convolutional neural networks," in *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*. ISCA, 2015, pp. 403–407.

[14] M. Jin, Y. Song, I. Mcloughlin, L. Dai, and Z. Ye, "Lid-senone extraction via deep neural networks for end-to-end language identification," in *Odyssey 2016: Speaker and Language Recognition Workshop, Bilbao, Spain, June 21-24, 2016*. ISCA, 2016, pp. 210–216.

[15] M. Jin, Y. Song, I. V. McLoughlin, W. Guo, and L. Dai, "End-to-end language identification using high-order utterance representation with bilinear pooling," in *INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*. ISCA, 2017, pp. 2571–2575.

[16] D. Garcia-Romero and A. McCree, "Stacked long-term TDNN for spoken language recognition," in *INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*. ISCA, 2016, pp. 3226–3230.

[17] M. Tkachenko, A. Yamshinin, N. Lyubimov, M. Kotov, and M. Nastasenko, "Language identification using time delay neural network d-vector on short utterances," in *Speech and Computer - 18th International Conference, SPECOM 2016, Budapest, Hungary, August 23-27, 2016, Proceedings*. Springer, 2016, pp. 443–449.

[18] J. Pesán, L. Burget, and J. Cernocký, "Sequence summarizing neural networks for spoken language recognition," in *INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*. ISCA, 2016, pp. 3285–3288.

[19] W. Geng, Y. Zhao, W. Wang, X. Cai, and B. Xu, "Gating recurrent enhanced memory neural networks on language identification," in *INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*. ISCA, 2016, pp. 3280–3284.

[20] J. Gonzalez-Dominguez, I. Lopez-Moreno, H. Sak, J. Gonzalez-Rodriguez, and P. J. Moreno, "Automatic language identification using long short-term memory recurrent neural networks," in *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*. ISCA, 2014, pp. 2155–2159.

[21] W. Geng, W. Wang, Y. Zhao, X. Cai, and B. Xu, "End-to-end language identification using attention-based recurrent neural networks," in *INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*. ISCA, 2016, pp. 2944–2948.

[22] G. Gelly and J. Gauvain, "Spoken language identification using lstm-based angular proximity," in *INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*. ISCA, 2017, pp. 2566–2570.

[23] R. Masumura, T. Asami, H. Masataki, and Y. Aono, "Parallel phonetically aware dnns and LSTM-RNNS for frame-by-frame discriminative modeling of spoken language identification," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*. IEEE, 2017, pp. 5260–5264.

[24] Z. Tang, D. Wang, Y. Chen, L. Li, and A. Abel, "Phonetic temporal neural model for language identification," *IEEE/ACM Trans. Audio, Speech & Language Processing*, vol. 26, no. 1, pp. 134–144, 2018.

[25] G. Gelly, J. Gauvain, V. B. Le, and A. Messaoudi, "A divide-and-conquer approach for language identification based on recurrent neural networks," in *INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*. ISCA, 2016, pp. 3231–3235.

[26] S. Fernando, V. Sethu, E. Ambikairajah, and J. Epps, "Bidirectional modelling for short duration language identification," in *INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*. ISCA, 2017, pp. 2809–2813.

[27] H. Zhao, D. Bansé, G. R. Doddington, C. S. Greenberg, J. M. Howard, J. Hernandez-Cordero, L. P. Mason, A. F. Martin, D. A. Reynolds, E. Singer, and A. Tong, "Results of the 2015 NIST language recognition evaluation," in *INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*. ISCA, 2016, pp. 3206–3210.

[28] "NIST 2017 language recognition evaluation plan."

[29] J. Nouza, R. Safarik, and P. Cerva, "ASR for south slavic languages developed in almost automated way," in *INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*. ISCA, 2016, pp. 3868–3872.

[30] "The 2015 NIST language recognition evaluation plan (LRE15)."

[31] R. Zazo, A. Lozano-Diez, and J. Gonzalez-Rodriguez, "Evaluation of an lstm-rnn system in different nist language recognition frameworks," in *Odyssey 2016: Speaker and Language Recognition Workshop, Bilbao, Spain, June 21-24, 2016*. ISCA, 2016, pp. 210–216.

Identification of Scandinavian Languages from Speech Using Bottleneck Features and X-vectors

Petr Cerva, Lukas Mateju, Frantisek Kynych, Jindrich Zdansky, and Jan Nouza

Institute of Information Technologies and Electronics,
Technical University of Liberec, Studentska 2, Liberec, 46117, Czech Republic

Abstract. This work deals with identification of the three main Scandinavian languages (Swedish, Danish and Norwegian) from spoken data. For this purpose, various state-of-the-art approaches are adopted, compared and combined, including i-vectors, deep neural networks (DNNs), bottleneck features (BTNs) as well as x-vectors. The best resulting approaches take advantage of multilingual BTNs and allow us to identify the target languages in speech segments lasting 5 seconds with a very low error rate around 1%. Therefore, they have many practical applications, such as in systems for transcription of Scandinavian TV and radio programs, where different persons speaking any of the target languages may occur. Within identification of Norwegian, we also focus on an unexplored sub-task of distinguishing between Bokmål and Nynorsk. Our results show that this problem is much harder to solve since these two language variants are acoustically very similar to each other: the best error rate achieved in this case is around 20%.

Keywords: spoken language identification, Scandinavian languages, x-vectors, bottleneck features, deep neural networks

1 Introduction

The Scandinavian category of languages is also referred to as North Germanic. It includes many related languages and dialects, from which the most widely spoken is, by a large margin, the triplet comprising Swedish, Danish and Norwegian. The first language, Swedish, is spoken by 9.2 mil. people and is written in a manner similar to Danish. However, Danish, a native language for 5.6 mil. people, sounds different from Swedish. The third language, Norwegian, spoken by 5.2 mil. people, shares many similarities with the first two languages so that a native Norwegian speaker is able to understand Danish as well as Swedish.

Interestingly, Norwegian has two standards, Bokmål ('Book Tongue') and Nynorsk ('New Norwegian'). The former standard is a Norwegianized variety of Danish and it is used by almost 85% of citizens in Norway. The latter variant is a language form based on Norwegian dialects (mostly from the west coast) and a puristic opposition to Danish. It is spoken by approximately 15% of the Norwegians.

There exist many differences between Bokmål and Nynorsk. For example, Bokmål has an optional feminine gender, favors noun-heavy expressions, heavily relies on prefixes like an- , be-, het- and the suffix -else, etc. On the contrary, Nynorsk has (more) Western diphthongs, and the feminine gender is compulsory for all feminine words; it favors verbal expressions over noun use, many abstract nouns are shorter, etc.

1.1 Motivation for this work

In our long-term project we have been developing a multi-lingual broadcast transcription and monitoring platform. One of its modules is devoted for spoken language identification. In this contribution, we focus on the identification of the three main Scandinavian languages. As mentioned above, they are mutually understandable – at least to a certain extent – and it often happens that, e.g., in a Norwegian TV program, a speaker using Swedish or Danish may appear. Some TV stations add subtitles in such situations, some do not. In radio broadcasting, subtitling is not possible at all. For automatic speech transcription, it is necessary to identify those parts of spoken content and assign them the proper language-specific modules. The same must also be done in other automated speech processing services used, e.g., in call centers.

Since these three languages are rather similar on the phonetic level, the identification task is not as easy as in the cases of more distant languages. Its complexity can be compared to, e.g., distinguishing between Czech and Slovak or Spanish and Catalan.

In case of Norwegian, we also investigate the possibility of distinguishing between Bokmål and Nynorsk. The hypothesis for this part of our work is that acoustic differentiation between them is possible, albeit only to a limited extent, on the basis of their origin and other slight differences in spelling and pronunciation. We focus on this sub-tasks since the existence of these two variants significantly complicates ASR of Norwegian due to the above-described lexical differences. The phenomenon of their mixing occurs even more often than for individual Scandinavian languages: each time Norwegian is transcribed, one of the standards must be chosen based on the prevailing features in the spoken content, or sometimes also on the speaker's preference.

2 State-of-the-art approaches to LID

Over the years, different advanced modeling techniques have been successfully applied to the task of language identification (LID). One of the more popular approaches is based on the total variability factor analysis; it is known as the i-vector framework [5]. An i-vector maps an utterance to a fixed-length representation. It jointly contains information about the speaker, language, and more. The computed i-vectors can be classified using, e.g., cosine scoring, multi-class logistic regression, or Gaussian models. The major drawback of the i-vectors is the degraded performance on shorter utterances [15].

Recently, deep learning has also been applied to LID in two major ways, indirect and direct. In the former case, BTNs have become the go-to state-of-the-art features widely used in many systems [21, 9] due to their superior performance. These features are usually first extracted from a DNN trained to discriminate individual physical states of a tied-state triphone model, and later used as inputs to either i-vector- [6] or DNN-based systems [10]. The training data (monolingual/multilingual) and the architecture of the BTN extractors (e.g., the placement of the bottleneck layer) have thoroughly been studied [7].

In the latter case, numerous end-to-end DNN-based systems have been proposed. Initially, they were trained to make a frame-level decision based on a frame-level input. At first, a probability vector (with a size corresponding to the number of the target

languages) is computed for each frame. After that, the probability vectors are averaged, and the class with the greatest value is chosen as the language spoken in the utterance. In [15], a feed-forward fully connected (FC) DNN outperformed the baseline i-vector system on short utterances. Since then, other complex architectures, including convolutional neural networks [16], time-delay neural networks (TDNNs) [10] or recurrent neural networks (RNNs) have been successfully adopted. The context modeling powers of RNNs have notably been exploited. Long short-term memory (LSTM) RNNs [14], gated recurrent unit RNNs [13] and bidirectional LSTM RNNs [8] have all yielded excellent results.

More recently, the focus of the DNNs has shifted from frame-level decisions to a single representation of the whole reference. In this case, the variable-length input utterance is mapped into a fixed-sized vector. After that, the vector can either be applied directly to produce the final decision or extracted and later used as an input to a classifier (analogically to the i-vectors). The mapping is usually done by integrating the statistics pooling layer [2] or learnable dictionary encoding [3]. Moreover, an attention mechanism has been proposed [12]. In [11], an angular proximity loss function was introduced.

Finally, the x-vector embeddings [23] yielded the best results in the 2017 NIST language recognition evaluation (LRE17) [22]. These embeddings, initially developed for speaker recognition [24], are able to encode various attributes of an input utterance including its length, channel information, speaker's gender, speaking rate or even spoken content [20]. They can be extracted using various DNN architectures with a temporal pooling layer and provide robust representations when a large amount of training data is used. The authors of [23] achieved the best results by using multilingual BTNs, data augmentation, and a discriminating Gaussian classifier.

3 Datasets and metrics

The dataset of Scandinavian languages used in this work consists of two parts. The first part, the train sub-set, contains 7 hours of speech utterances for every target language (i.e., for Bokmål, Nynorsk, Danish and Swedish). The second evaluation part consists of an additional 2,000 utterances (500 for every language) with an average length of 5 seconds. Note that all of the speakers represented in the training part of the data are different from those in the evaluation set. The acoustic channel is the same for all utterances and corresponds to broadcast news (the Norwegian data comes from the RUNDKAST database [1]).

For evaluation, the error rate (ER) and C_{avg} values are utilized. The first metric is defined as:

$$ER = \frac{F_{\mathrm{utt}}}{N_{\mathrm{utt}}} \ , \tag{1}$$

where $\mathrm{F_{utt}}$ is the number of falsely classified utterances, and $\mathrm{N_{utt}}$ is the total number of evaluated utterances. The latter metric is the official metric of the 2015 NIST Language Recognition Evaluation, C_{avg}. Detailed information about this closed set multi-language cost function and its definition can be found in the 2015 LRE Evaluation [26].

4 Baseline i-vector system and direct methods

To set a baseline performance, a 600-dimensional i-vector system has been trained using a full covariance GMM-UBM based system and logistic regression model. Within this training, MFCCs filtered by voice activity detection are employed. Note that this baseline approach follows the lre07 recipe, presented in Kaldi ASR toolkit [19].

The following three DNN architectures represent direct methods and they process the input speech segment frame-by-frame. That means that during the classification phase, a probability vector is obtained for each frame of given utterance (i.e., by doing a forward pass). These vectors are then averaged, and the language with maximum average probability is selected as an output.

4.1 Feed-forward FC DNN

The first architecture we adopt is a feed-forward FC DNN. This network was trained to directly distinguish between the target languages (i.e., direct method) by using a soft-max layer. During the training, the DNN hyper-parameters are set as follows: five hidden layers, 1,024 neurons per hidden layer, the rectified linear unit (ReLU) activation function, a learning rate of 0.08, and 20 training epochs. The network is trained over filter bank coefficients (FBCs) and the input feature vector is formed as a concatenation of 15 previous frames, current frame, and 15 following frames (i.e., 0.3-second context).

4.2 TDNN architecture

The next utilized architecture is represented by TDNN, which allows for capturing longer context in the input signal in a non-recursive way. It operates on frames with a temporal context centered on the current frame. The TDNN layers are built on top of the context of the earlier layers, and the final context is thus a sum of the partial ones.

Our trained TDNN also consists of five hidden layers, each with 1,024 neurons. The input context of each layer required to compute output at one time step includes three preceding inputs, the current input, and three following inputs (from the preceding layer). This setting matches the input context window size of the feed-forward DNN and the remaining hyper-parameters and input features are also unchanged.

4.3 FSMN architecture

Finally, vectorized feed-forward sequential memory networks (FSMNs) [25] are employed. This topology allows us to eliminate the recursion by adding several memory blocks with trainable weight coefficients into each layer of a standard feed-forward FC DNN. The memory blocks use a tapped-delay line structure to encode the long context information into a fixed-size representation.

This means that, in fact, FSMNs represent a generalization of TDNNs – the delayed inputs to each FSMN layer are weighted by the above-mentioned trainable matrix rather than concatenated in a fixed order as in the case of TDNNs. The important difference is that FSMNs in fact employ the sum pooling over time in order to limit the number

Table 1. Results of various LID approaches on a) the set of three Scandinavian languages and b) the extended set also distinguishing between Bokmål or Nynorsk (i.e., containing four languages).

	3 languages		**4 languages**	
	$ER[\%]$	$C_{avg}[\%]$	$ER[\%]$	$C_{avg}[\%]$
logistic regression + i-vectors	15.5	13.3	31.1	20.7
NN-based classifiers over FBCs				
DNN	16.0	10.4	31.3	20.9
TDNN	14.6	12.1	30.3	20.2
FSMN	17.9	13.1	32.2	21.5
TDNN classifier over BTNs				
BTNs-DNN-1	5.9	3.9	20.3	13.5
BTNs-FSMN-17	**0.7**	**0.1**	10.9	7.2
BTNs-stacked-17 from [7]	5.2	2.5	15.9	10.6
FSMN-based x-vectors with width of 512 neurons + DNN classifier				
over FBCs	3.7	2.7	22.9	15.2
over BTNs-FSMN-17	1.2	0.2	**10.5**	**7.0**

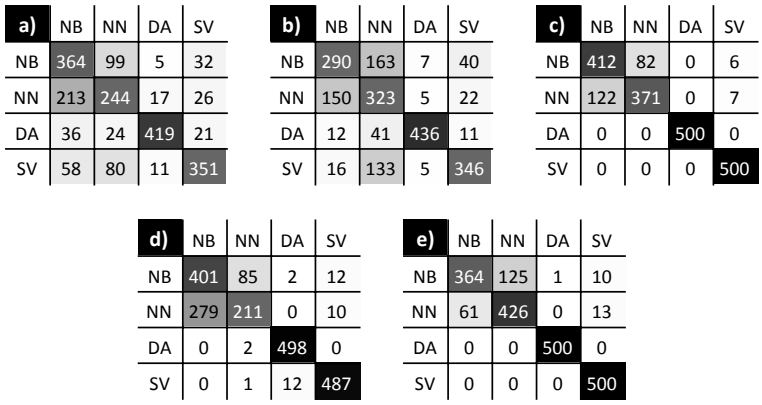
of trainable parameters, while TDNNs reduce the computation demands by using subsampling, which unfortunately limits the possibility of exploiting time dependencies in the input signal. To match the previous settings, the FSMN has 5 hidden layers, each with 1,024 neurons, and the context is the same as for the TDNN-based classifier.

4.4 Results

The results yielded by the i-vector system are presented in the first row of Table 1 and the results of three NN-based classifiers in its subsequent three rows. We can see that all these methods perform on a similar level and that the lowest ERs are yielded by the TDNN-based classifier. The results also show that the ERs calculated over the extended set including Bokmål or Nynorsk are approximately two times higher than for three languages (e.g., compare 30.3% ER with 14.6% ER for the TDNN classifier). This fact is also evident from the confusion matrices depicted in Fig. 1. Here, matrix a) corresponds to the i-vector system and matrix b) to the TDNN-based architecture, where most of the errors are caused by confusion between Bokmål and Nynorsk (and vice versa). The second most frequent source of errors for the TDNN classifier is Swedish, which is often incorrectly identified as Nynorsk. However, even the ER values achieved for the set of three languages are too high. In the next section, we try to reduce all these errors by employing BTN features.

5 BTN-based approaches

Two different types of bottleneck extractors are investigated. The first, monolingual DNN-based type, represents a baseline architecture. The second, advanced multilingual



a)	NB	NN	DA	SV
NB	364	99	5	32
NN	213	244	17	26
DA	36	24	419	21
SV	58	80	11	351

b)	NB	NN	DA	SV
NB	290	163	7	40
NN	150	323	5	22
DA	12	41	436	11
SV	16	133	5	346

c)	NB	NN	DA	SV
NB	412	82	0	6
NN	122	371	0	7
DA	0	0	500	0
SV	0	0	0	500

d)	NB	NN	DA	SV
NB	401	85	2	12
NN	279	211	0	10
DA	0	2	498	0
SV	0	1	12	487

e)	NB	NN	DA	SV
NB	364	125	1	10
NN	61	426	0	13
DA	0	0	500	0
SV	0	0	0	500

Fig. 1. Confusion matrices of different systems: a) i-vectors, b) TDNN-based classifier over FBCs, c) TDNN-based classifier over multilingual BTNs-FSMN-17, d) x-vectors over FBCs, and e) x-vectors over multilingual BTNs-FSMN-17. The abbreviations NB, NN, DA and SV stand for Bokmål, Nynorsk, Danish and Swedish, respectively.

FSMN-based extractor, yielded the best results in an extensive study focused on LID of Slavic languages in [4]. It also corresponds (except the BTN layer) to the architecture that we use to train the multilingual acoustic model for our ASR system. Note that, in both these cases, the BTN features form an input to the TDNN classifier, which yielded the best results in the previous section.

5.1 Monolingual feed-forward FC BTNs

The first conventional type of BTN extractor corresponds to a feed-forward FC topology with five hidden layers, the third one being the bottleneck layer. This DNN utilizes 1,024 neurons per hidden layer (39 for the bottleneck layer) and ReLU activation functions (sigmoid for the bottleneck layer). Their input is formed by 39 FBCs computed from 25 ms long frames with frame-shifts of 10 ms each. The DNN operates in the context of 11 frames (five preceding and five following the current frame). Normalization of the input feature vectors is performed within a 1-second window. These BTN features are trained as monolingual to discriminate between physical states (senones) of the tied-state tri-phone acoustic model of 48 Czech phonemes (including models of noises). The speech database used for the training contains 270 hours of Czech speech recordings. The resulting basic BTNs are further denoted as BTNs-DNN-1.

5.2 Multilingual FSMN-based BTNs

The more advanced FSMN-based extractor produces features denotes as BTNs-FSMN-17. Its input is again formed by 39 FBCs computed from 25-ms-long frames, but this time with frame-shifts of 12.5 ms each (the frame-shift is increased to speed up the speech recognition process in our current ASR system, where we utilize this multilingual topology for acoustic modelling). In this case 11 hidden layers are utilized, out of

which the tenth one is the BTN layer. The widths of common layers and of the BTN layer are 512 and 39 neurons, respectively. The context of each layer has nine elements, (four preceding and four following the current frame). The extractor is trained as multilingual for 17 languages (including, e.g., Swedish and Norwegian, English and 11 main Slavic languages) using block soft-max [7]. The training database contains 2,300 hours of clean speech and 240 hours of augmented speech data to improve the robustness to a) reverberation / noise [17] and b) telephone / speech codecs [18].

5.3 Results

Results of this experiment (see the third section in Table 1 show that even the basic monolingual BTNs significantly outperform the best TDNN-based classifier. For example, the ER for four languages is reduced from 31.3% to 20.3%. We can also see that the system with multilingual BTNs-FSMN-17 yields much better results than in the case of BTNs-DNN-1. At the same time, it allows us to distinguish between the triplet of languages with a very low ER of just 0.7% (see the confusion matrix c) in Fig. 1). We can also see that the identification of Bokmål and Nynorsk is also possible, but with a much higher ER of 20% (see the first two rows and columns of the confusion matrix).

For comparison, we also evaluated the extractor of stacked BTNs as presented in [7]. This extractor is trained by its authors for the same number of 17 languages as our FSMN-based extractor, but the representation of individual languages in its training set is different. The obtained results show that these 80-dimensional BTNs (denoted as BTNs-stacked-17 in Table 1) outperform the monolingual BTNs-DNN-1 but yield significantly worse results than our BTNs-FSMN-17.

6 The use of x-vectors

In this work, FSMN-based architecture is used for x-vector extraction. Its structure is described in Table 2, where the symbol ℓ denotes the current frame, on which the temporal context is centered. The pooling layer computes only the means of the frames (omitting the variances) in the context of 41 consecutive frames. In all neurons, the exponential linear unit (ELU) is used as the activation function.

On the input, each frame of the signal is represented by 39 FBCs computed from 25-ms-long frames with frame-shifts of 10 ms each. Table 2 shows that the extractor operates with a total context of 57 frames, which corresponds to 0.57 seconds. The parameter w stands for the dimensionality of the computed x-vectors. The x-vectors are extracted after the pooling layer. Note that, in training, we utilize a multiclass cross entropy objective function to classify $N_{languages}$.

6.1 Results

In the first experiment, cosine distance is used for classification and the x-vectors are extracted with a frame-shift of 41 frames (i.e., the context size of the pooling layer). The final x-vector representing the whole utterance is then obtained as an average of the shifted vectors. The results for different dimensionality of the x-vectors are presented

Table 2. The structure of FSMN-based x-vector extractor.

layer	**layer context**	**total context**	**input** x **output**
FSMN 1	$\ell \pm 4$	9	40×256
FSMN 2	$\ell \pm 4$	17	256×256
FC 1	ℓ	17	$256 \times w$
Pooling	$\ell \pm 20$	57	$(41 \cdot w) \times w$
FC 2	ℓ	57	$w \times w$
Softmax	$-$	57	$w \times N_{languages}$

Table 3. Results for the FSMN-based x-vectors with cosine distance on the 4-language set for different width of the pooling layer.

pooling layer's width	20	39	80	128	256	512	1024
$ER[\%]$	23.5	23.2	23.6	23.9	23.9	**22.9**	25.7

in Table 3. Here, the lowest ER of 22.9% has been achieved for the size equal to 512 (the other ERs are just slightly worse). However, this ER value is much higher than in the case of the BTNs.

To decrease it further, several different NN-based classifiers have been employed instead of cosine distance in the next experiment. These classifiers vary in the number of hidden layers used. The width of each layer is always 512 neurons. In this case, the x-vectors are extracted from the input utterance with shifts of one frame. The resulting sequence then forms the input to the classifier (i.e., the x-vectors are not averaged as for the use of the cosine distance). The results are presented in Table 4, where the lowest ER of 20.2% was achieved for 2 hidden layers. This value is by 2.7% lower than for the cosine distance.

The complete results for this setting (i.e., a width of 512 neurons and the DNN-based classier with two hidden layers) are shown on the penultimate row in Table 1 and correspond to the confusion matrix d) in Fig. 1. Here, we can see that the x-vectors over FBCs work well for three languages, but fail in the more difficult four-language task, where Nynorsk is in most cases misclassified as Bokmål.

The comparison of these results with the previous ones shows that x-vectors over FBCs outperform the i-vectors as well as the direct methods over the same features and achieve the ER value similar to basic monolingual BTNs. However, they are not able to outperform BTNs-FSMN-17, which benefits from a very large amount of the multilingual training data. Note that this data does not need to cover all target languages for identification (e.g., Danish is not included in our case).

In the last experiment, x-vectors are extracted using BTNs-FSMN-17 as input features. The results (see the last row in Table 1 and the matrix e) in Fig. 1) imply that this combination yields much better results than the extraction of x-vectors from FBCs. The results are also similar to the ones achieved by the TDNN-based classifier over the same features: the ER is slightly better for the 4-language set but slightly worse for the three languages (compare the sixth row with results in Table 1 with the last one).

Table 4. Results of the FSMN-based x-vectors with width of 512 neurons on the 4-language set for different NN-based classifiers

number of hidden layers	0	1	2	3	4	5
$ER[\%]$	22.7	20.5	**20.2**	20.3	21.3	21.7

7 Conclusions

In this paper, we have focused on the identification of four closely related Scandinavian languages, Swedish, Danish, and two Norwegian variants, Bokmål and Nynorsk. The experimental evaluation performed on short utterances (with an average length of 5 seconds) has shown that the direct NN-based classifiers trained on standard acoustic features (i.e., FBCs) perform comparably with the baseline i-vector-based system. A detailed analysis of the results in the form of confusion matrices has shown that all these systems generally make mistakes for all languages, with most errors occuring, as expected, between Bokmål and Nynorsk. The x-vectors extracted using FBCs have improved the results (except the ones between the Norwegian variants) in both ER and C_{avg} by better representing the utterances. Finally, the best results by a large margin are yielded by using multilingual BTNs as input features for the TDNN-based classifier or FSMN-based x-vector extractor. In these cases, the overall achieved ER around 1.0% for Swedish, Danish and Norwegian makes these approaches fully suitable for all applications where multiple Scandinavian languages may occur.

However, the great acoustic similarity between Bokmål and Nynorsk still causes ER around 20% for these two variants of Norwegian and opens a space for further research. A possible solution is to use an ASR with a mixed Bokmål and Nynorsk lexicon (and a language model) and perform the identification of these two variants later, based on their lexical features.

8 Acknowledgements

This work was supported by the Technology Agency of the Czech Republic (Project No. TO01000027), and by the Student Grant Competition of the Technical University of Liberec under project No. SGS-2019-3017.

References

1. Amdal, I., Strand, O.M., Almberg, J., Svendsen, T.: RUNDKAST: an annotated norwegian broadcast news speech corpus. In: LREC 2008, Marrakech, Morocco. pp. 1907–1913 (2008)
2. Cai, W., Cai, Z., Liu, W., Wang, X., Li, M.: Insights in-to-end learning scheme for language identification. In: ICASSP 2018, Calgary, AB, Canada. pp. 5209–5213 (2018)
3. Cai, W., Cai, Z., Zhang, X., Wang, X., Li, M.: A novel learnable dictionary encoding layer for end-to-end language identification. In: ICASSP 2018, Calgary, AB, Canada. pp. 5189–5193 (2018)

4. Cerva, P., Mateju, L., Zdansky, J., Safarik, R., Nouza, J.: Identification of related languages from spoken data: Moving from off-line to on-line scenario. Computer Speech & Language **68** (2021)
5. Dehak, N., Torres-Carrasquillo, P.A., Reynolds, D.A., Dehak, R.: Language recognition via i-vectors and dimensionality reduction. In: Interspeech 2011, Florence, Italy. pp. 857–860 (2011)
6. Fer, R., Matejka, P., Grezl, F., Plchot, O., Cernocky, J.: Multilingual bottleneck features for language recognition. In: Interspeech 2015, Dresden, Germany. pp. 389–393 (2015)
7. Fer, R., Matejka, P., Grezl, F., Plchot, O., Vesely, K., Cernocky, J.H.: Multilingually trained bottleneck features in spoken language recognition. Compututer Speech & Language **46**, 252–267 (2017)
8. Fernando, S., Sethu, V., Ambikairajah, E., Epps, J.: Bidirectional modelling for short duration language identification. In: Interspeech 2017, Stockholm, Sweden. pp. 2809–2813 (2017)
9. Ferrer, L., Lei, Y., McLaren, M., Scheffer, N.: Study of senone-based deep neural network approaches for spoken language recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing **24**(1), 105–116 (2016)
10. Garcia-Romero, D., McCree, A.: Stacked long-term TDNN for spoken language recognition. In: Interspeech 2016, San Francisco, CA, USA. pp. 3226–3230 (2016)
11. Gelly, G., Gauvain, J.: Spoken language identification using lstm-based angular proximity. In: Interspeech 2017, Stockholm, Sweden. pp. 2566–2570 (2017)
12. Geng, W., Wang, W., Zhao, Y., Cai, X., Xu, B.: End-to-end language identification using attention-based recurrent neural networks. In: Interspeech 2016, San Francisco, CA, USA. pp. 2944–2948 (2016)
13. Geng, W., Zhao, Y., Wang, W., Cai, X., Xu, B.: Gating recurrent enhanced memory neural networks on language identification. In: Interspeech 2016, San Francisco, CA, USA. pp. 3280–3284 (2016)
14. Gonzalez-Dominguez, J., Lopez-Moreno, I., Sak, H., Gonzalez-Rodriguez, J., Moreno, P.J.: Automatic language identification using long short-term memory recurrent neural networks. In: Interspeech 2014, Singapore, September 14-18, 2014. pp. 2155–2159 (2014)
15. Lopez-Moreno, I., Gonzalez-Dominguez, J., Plchot, O., Martinez, D., Gonzalez-Rodriguez, J., Moreno, P.J.: Automatic language identification using deep neural networks. In: ICASSP 2014, Florence, Italy. pp. 5337–5341 (2014)
16. Lozano-Diez, A., Zazo-Candil, R., Gonzalez-Dominguez, J., Toledano, D.T., Gonzalez-Rodriguez, J.: An end-to-end approach to language identification in short utterances using convolutional neural networks. In: Interspeech 2015, Dresden, Germany. pp. 403–407 (2015)
17. Malek, J., Zdansky, J.: On practical aspects of multi-condition training based on augmentation for reverberation-/noise-robust speech recognition. In: TSD 2019, Ljubljana, Slovenia. pp. 251–263 (2019)
18. Malek, J., Zdansky, J., Cerva, P.: Robust recognition of conversational telephone speech via multi-condition training and data augmentation. In: TSD 2018, Brno, Czech Republic. pp. 324–333 (2018)
19. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., Vesely, K.: The Kaldi speech recognition toolkit. In: ASRU 2011, Waikoloa, HI, USA (2011)
20. Raj, D., Snyder, D., Povey, D., Khudanpur, S.: Probing the information encoded in x-vectors. In: ASRU 2019, Singapore. pp. 726–733 (2019)
21. Richardson, F., Reynolds, D.A., Dehak, N.: Deep neural network approaches to speaker and language recognition. IEEE Signal Processing Letters **22**(10), 1671–1675 (2015)

22. Sadjadi, S.O., Kheyrkhah, T., Tong, A., Greenberg, C.S., Reynolds, D.A., Singer, E., Mason, L.P., Hernandez-Cordero, J.: The 2017 NIST language recognition evaluation. In: Odyssey 2018, Les Sables d'Olonne, France. pp. 82–89 (2018)
23. Snyder, D., Garcia-Romero, D., McCree, A., Sell, G., Povey, D., Khudanpur, S.: Spoken language recognition using x-vectors. In: Odyssey 2018, Les Sables d'Olonne, France. pp. 105–111 (2018)
24. Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., Khudanpur, S.: X-vectors: Robust DNN embeddings for speaker recognition. In: ICASSP 2018, Calgary, AB, Canada. pp. 5329–5333 (2018)
25. Zhang, S., Liu, C., Jiang, H., Wei, S., Dai, L., Hu, Y.: Feedforward sequential memory networks: A new structure to learn long-term dependency. CoRR **abs/1512.08301** (2015)
26. Zhao, H., Banse, D., Doddington, G.R., Greenberg, C.S., Hernandez-Cordero, J., Howard, J.M., Mason, L.P., Martin, A.F., Reynolds, D.A., Singer, E., Tong, A.: Results of the 2015 NIST language recognition evaluation. In: Interspeech 2016, San Francisco, CA, USA. pp. 3206–3210 (2016)

Computer Speech & Language 68 (2021) 101180

Contents lists available at ScienceDirect

Computer Speech & Language

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

Identification of related languages from spoken data: Moving from off-line to on-line scenario

Petr Cerva*, Lukas Mateju, Jindrich Zdansky, Radek Safarik, Jan Nouza

Faculty of Mechatronics, Informatics and Interdisciplinary Studies, Technical University of Liberec, Studentska 2, Liberec 461 17, Czech Republic

ARTICLE INFO

Article History:
Received 24 August 2020
Revised 3 December 2020
Accepted 11 December 2020
Available online 15 December 2020

Keywords:
Spoken language identification
Deep neural networks
Weighted finite-state transducers
On-line processing
Slavic languages

ABSTRACT

The accelerating flow of information we encounter around the world today makes many companies deploy speech recognition systems that, to an ever-growing extent, process data on-line rather than off-line. These systems, e.g., for real-time 24/7 broadcast transcription, often work with input-stream data containing utterances in more than one language. This multilingual data can correctly be transcribed in real-time only if the language used is identified with just a small latency for each input frame. For this purpose, a novel approach to on-line spoken language identification is proposed in this work. Its development is documented within a series of consecutive experiments starting in the off-line mode for 11 Slavic languages, going through artificially prepared multilingual data for the on-line scenario, and ending with real bilingual TV programs containing utterances in mutually similar Czech and Slovak. The resulting scheme that we propose operates frame-by-frame; it takes in a multilingual stream of speech frames and outputs a stream of the corresponding language labels. It utilizes a weighted finite-state transducer as a decoder, which smooths the output from a language classifier fed by multilingual and augmented bottleneck features. An essential factor from the accuracy point of view is that these features, as well as the classifier itself, are based on deep neural network architectures that allow the modeling of long-term time dependencies. The obtained results show that our scheme allows us to determine the language spoken in real-world bilingual TV shows with an average latency of around 2.5 seconds and with an increase in word error rate by a mere 2.9% over the reference 18.1% value yielded by using manually prepared language labels.

1. Introduction

The goal of spoken language identification (SLI) is to correctly determine the language spoken in a speech utterance (Li et al., 2013). In the recent years, extensive scientific efforts have been dedicated to this task. For example, eight NIST language recognition evaluations (web, 2020), focusing mainly on telephone conversations, were conducted from 1996 to 2017.

Nowadays, SLI systems form an integral part of many speech-processing applications including, e.g., systems for spoken language translation or multilingual speech recognition. SLI systems are also used in dialog systems (Griol et al., 2020), emergency call routing or in solutions for spoken document retrieval and indexing. The possibility of automatic identification or verification of the language spoken by a given person is also very important in the area of security and intelligence.

*Corresponding author.
E-mail addresses: petr.cerva@tul.cz (P. Cerva), lukas.mateju@tul.cz (L. Mateju), jindrich.zdansky@tul.cz (J. Zdansky), radek.safarik@tul.cz (R. Safarik), jan.nouza@tul.cz (J. Nouza).

https://doi.org/10.1016/j.csl.2020.101180
0885-2308/

Basically, there exist two main possibilities for performing SLI: in a) off-line or b) on-line (real-time) mode. The input to the former and classic scenario is usually formed by one speech recording or utterance. Its entire content, i.e., all speech samples or frames, may then be utilized for classification. Moreover, multiple recognition passes are sometimes performed in this case as the required computation time is usually not strictly limited.

But today's world is accelerating; the data processing and information mining domains face a new challenge when their users ask for very quick results and analysis, ideally during the data flow. Moving from off-line to on-line scenario is also important in automatic speech recognition (ASR) tasks, such as continual (24/7) broadcast monitoring, real-time subtitling, on-line meeting transcriptions, debates, etc.

In this case, an on-line SLI system, which operates differently from an off-line one, must be employed: it has to be able to take in a multilingual sequence (stream) of speech frames on its input, and provide a stream of labels identifying the language spoken in every frame on its output. In addition, the practically deployable system must work well not only for multilingual, but also for monolingual data streams, where it must not degrade the transcription results.

Therefore, the requirements on the properties of on-line SLI systems are different and much more demanding. Typically, in many real-time applications, the most critical factor is that the decision about the language spoken has to be made with only a small delay (latency). This means that

- the computational demands of an on-line SLI approach have to be low;
- it may utilize only a limited number of surrounding frames for classification of an input speech frame;
- only one left-to-right pass through the input data can be performed.

Unfortunately, these restrictions are usually not taken into account during the design of most off-line SLI methods, and their usability for on-line mode is therefore limited (or not discussed in the respective papers). That also means that the number of approaches existing for real-time processing (e.g., of news broadcasting) is limited (see also a survey of the state-of-the-art methods in Section 2.2).

1.1. Motivation and contribution of this work

As mentioned above, media monitoring is one of the typical applications where streamed data is processed. For example, the main utilization of our automatic speech recognition technology is within a cloud platform for real-time transcription of TV and radio stations in several languages including Czech, Slovak, Polish and other, predominantly Slavic, languages (see our multilingual radio monitoring application[1]).

The automatic transcripts are further used by a media monitoring company in two different services. The first, broadcast monitoring and archiving, produces exact verbatim transcriptions (based on the manually corrected automatic ones) for a large set of selected programs (e.g., approximately 920 programs on a weekly basis for Czech). Within the scope of the second service, called TV/R Alerts, the results are sent to clients without corrections and with a minimum delay after the broadcasting time. The service operates in a 24/7 mode for all relevant TV and radio streams (74 stations for Czech, 21 stations for Slovak, etc).

Unfortunately, some of the target programs for both of these services contain utterances belonging to more than one language. For example, English is sometimes present and accompanied with subtitles in many news programs. However, in our case, the worst obstacle follows from Czech and Slovak often occurring side by side in many programs of many TV/R stations. This is namely true for programs such as interviews or talk-shows, where the invited person or the presenter may not speak the same language and, at the same time, the amounts of utterances pronounced in both languages are similar.

The reasons why Czech and Slovak may occur in one TV/R program are as follows: the Czech Republic and Slovakia formed one country called Czechoslovakia from 1918 to 1992; hence many people born in one country now live in the other one, and both languages are similar to the extent that Czech native speakers can understand Slovak and vice versa.

From the ASR point of view, these two languages have almost the same phonetic inventory (Slovak has several extra phonemes, but they are all similar to phonemes also existing in Czech), but differ in more than 75% of their lexical inventories. That means that it is impossible to transcribe a recording belonging to one language with a system trained for the other one.

A very similar situation also occurs in former Yugoslavia's countries, namely in Croatia, Serbia and Slovenia, and in many post-Soviet states. The language mixing phenomenon frequently occurs also in some Scandinavian countries (Sweden, Norway and Denmark) whose languages are at least partly mutually intelligible, too.

Therefore, the real-time SLI scenario is the main subject of our interest. The main contribution of our work is a new approach to on-line SLI that is capable of processing the multilingual (or bilingual) input data stream frame-by-frame with an average latency within 3 seconds and with just a minimum increase in word error rate (WER) over the reference value.

The development of the approach starts in the off-line mode. Here, the dataset used consists of recordings of the 11 most widespread Slavic languages. It corresponds to the target domain of our ASR system and is challenging as a lot of the Slavic languages are related to each other and hard to distinguish between as shown for example in Abdullah et al. (2020). This is especially true for those pairs of languages that belong to the same language branch, such as Czech and Slovak (West Slavic branch) as we

[1] https://tul-speechlab.gitlab.io/

have already mentioned. Other examples include Croatian, Serbian and Slovene (South Slavic branch), or Russian, Ukrainian and Belorussian (East Slavic Branch) which are difficult to distinguish even for native speakers.

To find out the best method for identification of these languages, various state-of-the-art deep neural network (DNN) based techniques are investigated and compared. The obtained results imply that the most important factor for a proper classification is the use of bottleneck (BTN) features that are extracted using the feed-forward sequential memory network (FSMN) architecture. Moreover, the BTN extractor is also trained as multilingual and using augmented data. Note that this off-line evaluation represents an extension of our previous work (Mateju et al., 2018).

After that, our attention moves to the on-line mode. Here, a weighted finite-state transducer (WFST) is proposed as a decoder, which smooths the output from the language recognizer under the hood. The whole scheme is evaluated on a multilingual dataset prepared artificially by combining sequences of recordings of all 11 Slavic languages at first. After that, a much more challenging development set is utilized to analyze and further tune the performance for the target bilingual task. It consists of real talk-shows containing utterances in Czech and Slovak.

Finally, utilizing the experience from all the previous experiments, the proposed scheme with the best parameter settings is evaluated on a test set compiled from real recordings of bilingual interviews.

Our additional contribution to the speech research community is a dataset of spoken Slavic recordings for off-line as well as on-line evaluation (including all the necessary annotations); we have made this dataset public.[2] Its part has been already used by other authors for a similar task, recently (Abdullah et al., 2020).

2. State of the art

As already stated in the introductory section, there are two main approaches to operating a standard spoken language recognizer: in off-line or on-line mode. In the literature, most of the proposed methods fully focus on the former mode while the latter, more restricted one, is covered very rarely. This section thus reviews the current state of the art in SLI for both of these modes.

2.1. Off-line mode

In recent years, many advanced acoustic modeling techniques have been applied to the task of off-line SLI. One of the most popular approaches, known as the i-vector framework (Dehak et al., 2011; Gonzalez et al., 2011), is based on the total variability factor analysis. An i-vector is a fixed-length representation of an utterance. It contains aggregate information about the speaker, language, and more (e.g., LDA can be used to obtain discriminating features). To extract the i-vectors, hand-crafted shifted delta cepstral features (SDCs) derived from mel-frequency cepstral coefficients (MFCCs) (Singer et al., 2012) and phone log-likelihood ratios (PLLRs) (D'Haro et al., 2014) have been the most commonly used input features. After this extraction, the computed i-vectors can be compared with the reference ones and classified using, e.g., multi-class logistic regression, cosine scoring, or Gaussian models. The primary issue of the i-vector framework is the degraded performance on shorter test utterances (Lopez-Moreno et al., 2014).

Recently, similar to all other speech processing tasks (Siniscalchi et al., 2014; Liu et al., 2017) (e.g., speech recognition Dahl et al., 2012), deep neural networks have established a dominating role and led to improved performance of SLI. There are two main approaches to SLI utilizing deep learning: indirect and direct ones; combinations of both have been popular as well. In the former case, bottlenecks have become the preferred state-of-the-art features widely used in many SLI systems (McLaren et al., 2016; Ferrer et al., 2016; Richardson et al., 2015b; 2015a) due to their superior performance. These features are usually extracted from a DNN trained to discriminate individual physical states of a tied-state triphone model. After that, they are further used as inputs to either i-vector-based (Song et al., 2015; Fer et al., 2015) or DNN-based systems (e.g., Garcia-Romero and McCree, 2016). The training data (monolingual vs. multilingual) and the architecture of the bottleneck extractors, including, e.g., the placement of the bottleneck layer, have extensively been studied (e.g., Fer et al., 2017).

In the latter approach, various end-to-end systems utilizing different DNN architectures have been trained to directly detect the language spoken in the utterance. At first, the DNNs were trained to produce a frame-level decision based on a frame-level input. In this case, the DNNs are set to compute a probability vector (of a size corresponding to the number of the target languages) for each frame. After that, the probability vectors are averaged, and the language with the maximum value is chosen as the output one. In Lopez-Moreno et al. (2014), the authors employed fully connected feed-forward DNNs for SLI on short utterances, outperforming the baseline i-vector system. Since then, more complex architectures, such as convolutional neural networks (CNNs) (Lozano-Diez et al., 2015), time-delay neural networks (TDNNs) (Garcia-Romero and McCree, 2016) or recurrent neural networks (RNNs), have been successfully applied.The context modeling ability of gated recurrent unit (GRU) RNNs (Geng et al., 2016b), long short-term memory (LSTM) RNNs (Gonzalez-Dominguez et al., 2014) and bidirectional (B) LSTM RNNs (Gelly et al., 2016) have especially been exploited for SLI. In Fernando et al. (2017), the BLSTM RNNs were applied on short utterances showing supreme performance over the i-vector approach.

More recently, a shift from frame-level decisions to a single representation of the whole reference has occurred. In this case, the variable-length speech input is mapped into a fixed-sized vector, which either directly determines the final decision; or, alternatively, it is extracted and later fed into a classifier (thus mimicking the i-vector framework). This mapping is usually

[2] https://owncloud.cesnet.cz/index.php/s/gXHKFs9UDEqe34G

performed by integrating a pooling mechanism into the training DNN. For example, statistics pooling layer (Cai et al., 2018a; Mingote et al., 2019), spatial pyramid pooling (Jin et al., 2018), self-attentive pooling (Cai et al., 2018c) or learnable dictionary encoding (Cai et al., 2018b) can be used. Additionally, an attention mechanism has successfully been used in DNNs (V. et al., 2016), RNNs (Geng et al., 2016a; Padi et al., 2019b; 2019a), and CNN-BLSTM (Cai et al., 2019). In Wan et al. (2019), the authors proposed a tuple-max loss function designed for classification tasks where the decision is restricted. The angular proximity loss function was introduced in Gelly and Gauvain (2017). Sequence summarizing neural networks were also explored in Pesan et al. (2016). BLSTM with a sequence summarizing layer was used to extract embeddings in Lozano-Diez et al. (2018). Finally, the embeddings called x-vectors (Snyder et al., 2018a), originally developed for speaker recognition (Snyder et al., 2018b), yielded the best results in the last NIST language recognition evaluation (LRE17) (LRE, 2017). These embeddings are produced by a DNN with a temporal (statistics) pooling layer. The extracted x-vectors can then be used in the same way as i-vectors. The authors of Snyder et al. (2018a) achieved the best results by using multilingual bottleneck features, data augmentation, and a discriminating Gaussian classifier. Since then, several works continued in further exploring the x-vectors (Lopez et al., 2018; Miao et al., 2019).

2.2. On-line processing

So far, all of the cited approaches have aimed at the best spoken language identification performance possible, and, naturally, all of them are applicable in the off-line mode. However, the restrictions for the on-line mode listed in the introductory section are often not considered in such designs, and the usability of these methods for this scenario is therefore limited (or not discussed in the cited papers). The limited amount of works that consider on-line processing include the vector space modeling (Li et al., 2007) and phone recognition followed by language modeling (PRLM) (Caseiro and Trancoso, 1998). Parallel PRLM was explored and compared with Gaussian mixture models (GMMs) and PRLM in Zissman (1996). Alternatively, phone lattices were used in Gauvain et al. (2004). In Lim et al. (2010), the authors proposed a joint language identification and speech recognition that can operate in near real-time. In their work, partial hypotheses generated during the decoding stages are compared, and the language decision is obtained after the first full hypothesis is generated. In Okamoto et al. (2017), the authors presented a method to reduce the latency of the speech-recognition-based spoken language recognizer by applying variable timeouts for each decoder. Finally, parallel phonetically aware DNNs and LSTM RNNs were proposed to enhance the frame-level SLI in Masumura et al. (2017). The authors also discussed the utilization of their approach in real-time.

3. Investigated and adopted bottleneck features

As mentioned in the previous section, it has been shown that BTN features outperform conventional features, such as MFCCs or filter bank coefficients (FBCs), in the SLI task (Fer et al., 2017). Therefore, in this work, namely BTN features were employed for classification. Specifically, three different types were investigated, all derived from neural network architectures designed for training of our ASR system.

3.1. Feed-forward fully connected BTNs

The first conventional type corresponds to a feed-forward fully connected topology with five hidden layers, the third one being the bottleneck layer. This neural network utilizes 1024 neurons per hidden layer (39 for the bottleneck layer) and rectified linear unit (ReLU) activation functions (sigmoid for the bottleneck layer). Their input is formed by 39 FBCs computed from 25 ms long frames with frame-shifts of 10 ms each. The neural network (NN) operates in the context of 11 frames (five preceding and five following the current frame). Normalization of the input feature vectors is performed within a 1-second window. Thus, the latency of this architecture, given by one half of the normalization window, is 500 ms.

These BTN features are trained as monolingual as well as multilingual tools to discriminate between physical states (senones) of tied-state tri-phone acoustic models. In the first basic case, a set of 48 Czech phonemes (including models of noises) and 270 h of Czech speech recordings are utilized. The resulting BTNs are further denoted as DNN-1l-48ph.

In the second scenario, 220 total hours of speech data belonging to 11 Slavic languages are employed, i.e., 20 h per every language. This training is carried out using four different sets of phonemes formed by grouping of all possible phonemes of all 11 target Slavic languages (based on expert knowledge of their phonetic similarity). The largest set of phonemes contains 70 units, the smallest just 14. These BTNs are denoted as DNN-11l-70ph, DNN-11l-36ph, DNN-11l-27ph and DNN-11l-14ph.

3.2. Context modeling architectures

The next investigated networks allow us to model a larger time context. Nowadays, this is mainly solved by recurrent networks and their variants such as LSTM or BLSTM – cf. the preceding section. However, the recurrent architectures also have some drawbacks; namely, a more complex and time consuming training. The RNNs use an infinite (in principle) recursion to model time dependency. In this paper, two approaches are adopted that allow for capturing the context in a non-recursive way.

3.2.1. TDNN-based BTNs

The first is the already mentioned TDNN topology, which operates on frames with a temporal context centered on the current frame. The TDNN layers are built on top of the context of the earlier layers, thus the final context is a sum of the partial ones.

Using the TDNN architecture, the ASR accuracy is improved with an increasing number of hidden layers better than for the feed-forward fully connected topology. Therefore, 11 hidden layers are utilized in this case, out of which the tenth one is the bottleneck layer. This layer is the only one with a sigmoid activation function, while exponential linear units (ELUs) are used for the remaining layers. The width of each layer is 512 neurons except for the BTN layer, which has just 39 neurons. The context of each layer is nine frames (four preceding and four following the current frame). The training of the network converges best when sub-sampling has been employed. The best results are yielded for context {-4,-2,0,-2,4}.

The input to this architecture is again formed by 39 FBCs computed from 25-ms long frames with a frame-shift of 10 ms each. The normalization of the input feature vectors is the same, and the built-in latency is 500 ms, too.

Only the multilingual training is applied to TDNNs. We have used the set of 27 phonemes as it yielded the best results in one part of the off-line evaluation (see Table 4). The resulting BTNs are denoted as TDNN-11l-27ph and their performance is directly comparable with the one yielded by the DNN-11l-27ph architecture.

3.2.2. FSMN-based BTNs

Finally, the architecture based on vectorized feed-forward sequential memory networks (Zhang et al., 2015) is employed. This topology allows to eliminate the recursion by adding several memory blocks with trainable weight coefficients into each layer of a standard feed-forward fully connected DNN. The memory blocks use a tapped-delay line structure to encode the long context information into a fixed-size representation.

This means that, in fact, FSMNs represent a generalization of TDNNs – the delayed inputs to each FSMN layer are weighted by the above-mentioned trainable matrix rather than concatenated in a fixed order as in the case of TDNNs. The important difference is that FSMNs employ in fact sum pooling over time in order to limit the number of trainable parameters while TDNNs reduce the necessary computation demands by using sub-sampling, which unfortunately limits the possibility of exploiting time dependencies in the input signal (sub-sampling is based on the assumption that neighboring activations are correlated).

The input to the FSMN architecture is again formed by 39 FBCs computed from 25-ms-long frames, but this time with frame-shifts of 12.5 ms each. As in the previous case, the same number of 11 hidden layers are utilized, out of which the tenth one is the bottleneck layer. The widths of common layers and of the bottleneck layer are 512 and 39 neurons, respectively. The context of each layer is again nine (four preceding and four following the current frame).

In this case, no normalization of the input feature vectors needs to be performed. Therefore, the latency of this topology is 11 (number of layers) x 4 (half of the context) x 12.5 ms (frame-shift), i.e., 550 ms, which is a value just slightly higher than for the previous systems.

At first, similar to the TDNN-based BTNs, only the multilingual training is performed. The resulting BTNs are denoted as FSMN-11l-27ph. Second, not only 20 h of speech data for every language, but all available data for 11 Slavic language are used, i.e., 1850 h in total. Moreover, block soft-max has been employed rather than phoneme grouping because it should yield better performance (Fer et al., 2017). That means that a specific output layer corresponding to the target language is activated for every input frame during the training. The corresponding BTNs are marked as FSMN-11l-BS+data. Finally, 240 h of augmented speech data are added to the training database to improve the robustness to a) reverberation/noise (Malek and Zdansky, 2019) and b) telephone/speech codecs (Malek et al., 2018). These last BTNs are marked as FSMN-11l-BS+data+aug.

4. Investigation and experimental evaluation in off-line mode

Within this section, we present our neural network approach to SLI targeted for off-line use. First, we introduce our dataset of Slavic language recordings, list the evaluation metrics, and show the results of a baseline i-vector-based system. This is followed up by an extensive evaluation of different neural network architectures and their influence on the performance of SLI. Finally, several bottleneck features are explored, and comprehensive error analysis is performed. Please note that this section is an extension of our previous work published in Mateju et al. (2018).

4.1. Dataset of Slavic language recordings

Slavic languages are spoken by approximately 320 million people, mostly across Central, Eastern, and Southern Europe, as well as the northern parts of Asia. There are more than ten languages with more than a million native speakers (e.g., Russian ($\sim$ 150 million speakers), Polish ($\sim$ 50 million), Ukrainian, etc.).

Slavic languages are classified into three main branches based on geographical and genealogical principles: East, South, and West Slavic languages. Most of the languages belonging to the same branch are close to each other and share many similarities. For example, Croatian and Serbian are practically identical in their phonetics. However, some languages may have different phonetics (e.g., Bulgarian or Polish are phonetically more similar to East Slavic languages than to the languages in their own branches). Every Slavic language has its unique phonetic inventory, which distinguishes it from the other languages (with the exception of the previously mentioned Croatian and Serbian). This information can be exploited in SLI. Moreover, rich morphology, a high degree of inflection, and more or less free word order result in a large linguistic complexity of all these languages.

Table 1
Overview of Slavic languages used.

Language	Country code	Branch	Number of speakers [millions]
Belorussian	BY	East Slavic	7.5
Russian	RU	East Slavic	150
Ukrainian	UA	East Slavic	40
Bulgarian	BG	South Slavic	8
Croatian	HR	South Slavic	5.6
Macedonian	MK	South Slavic	3
Serbian	RS	South Slavic	12
Slovene	SI	South Slavic	2
Czech	CZ	West Slavic	13.2
Polish	PL	West Slavic	50
Slovak	SK	West Slavic	7

In our previous work (Mateju et al., 2018), we released to the general public a first (and only) extensive audio dataset covering a large proportion of all Slavic languages for the task of SLI. In total, this dataset covers the 11 most widespread Slavic languages from all three main language branches as summarized in Table 1:

The dataset contains mostly recordings of clean speech as their original purpose was acoustic-model training for automatic speech recognition. The majority of these recordings come from TV and radio broadcasts, and the process of retrieving this data is described in detail in Nouza et al. (2016). The rest of the dataset consists of microphone recordings.

For each language, the compiled data is reasonably diverse (i.e., multiple speakers of both genders are represented, with several TV/R or microphone sources, and shorter and longer recordings for training). The dataset is divided into two non-overlapping subsets: 20 h of training data and 500 utterances for evaluation are available per language. For the evaluation phase, the focus is on shorter recordings (similar to Lopez-Moreno et al., 2014). For this reason, the maximum duration of an evaluation utterance is five seconds.

4.2. Evaluation metrics

To evaluate the performance of SLI in an off-line setting, two distinct metrics, namely, error rate (ER) and C_{avg}, are utilized.

The first metric, error rate, is defined as:

$$ER = \frac{F_{\text{utt}}}{N_{\text{utt}}} \quad , \tag{1}$$

where F_{utt} is the number of falsely classified utterances, and N_{utt} is the total number of evaluated utterances.

The official metric of the 2015 NIST Language Recognition Evaluation, C_{avg}, is the second metric. Detailed information about this closed set multi-language cost function and its definition can be found in the 2015 LRE Plan (LRE, 2015).

4.3. Baseline i-vector approach

As a baseline, an i-vector-based system was trained using a full covariance GMM-UBM-based system and the logistic regression model. It utilizes MFCCs filtered by voice activity detection as features, and it produces 600-dimensional i-vectors. Note that this baseline approach follows the lre07 v1 recipe available in the Kaldi ASR toolkit[3] (Povey et al., 2011).

The results of this i-vector-based system are summarized in the first row of Table 2. These results (i.e., ER of 4.2% and C_{avg} of 2.3%) provide a good starting point for subsequent experimental evaluation.

Moreover, we have also followed the v2 version of the Kaldi lre07 recipe. In this case, the GMM-UBM portion of the system is replaced by a TDNN-based UBM. The TDNN was trained on 270 hours of Czech speech (i.e., using the same data as DNN-1l-48ph extractor; see Section 3.1 for more information).

The results are presented in the second row of Table 2. This improved i-vector-based system yields a slight reduction in both metrics (e.g., by 0.3% in ER).

4.4. Investigated neural network architectures for classification

As already stated, deep neural networks play a role of a key component in our spoken language recognizer. In this section, we explore different neural network architectures and their best usage as classifiers for SLI. Architectures such as fully connected deep neural networks, convolutional neural networks, recurrent neural networks and time-delay neural networks, are tested. Note that the hyper-parameters of the networks are derived from our previous work (Mateju et al., 2018) and experiments we conducted prior to it. All the networks were trained and evaluated using the Slavic dataset.

[3] http://kaldi-asr.org/

Table 2
Results of baseline i-vector systems obtained for 11 languages and 500 utterances per language.

Approach	ER [%]	C_{avg} [%]
logistic regression + i-vectors	4.2	2.3
logistic regression + i-vectors + TDNN	3.9	2.1

4.4.1. Feed-forward fully connected DNN architecture

The first architecture we adopt is a feed-forward fully connected DNN. This network was trained to directly distinguish between 11 Slavic languages (i.e., direct method). For this reason, its output was formed by a soft-max layer with 11 neurons. During the training, the DNN hyper-parameters were set as follows: five hidden layers, 1024 neurons per hidden layer, the ReLU activation function, a learning rate of 0.08, and 20 training epochs. As for the features, both standard FBCs and DNN-1l-48ph BTNs were utilized. Finally, the input feature vector was formed as a concatenation of 15 previous frames, current frame, and 15 following frames (i.e., 0.3-second context). Note that different context sizes were studied in our previous work (Mateju et al., 2018).

During the classification phase, a probability vector was obtained for each frame of the given utterance (i.e., by doing a forward pass). After that, the probability vectors were averaged, and the language with the maximum average probability was selected as the output.

The obtained results are, for FBCs and DNN-1l-48ph BTNs, summarized in the first row of Table 3. They show that the standard FBCs do not exceed the baseline i-vector-based system (0.6% difference in ER). On the contrary, BTNs significantly outperform the baseline system (even the improved one). The improvements are 3% in ER (from 4.2% to 1.2%) and more than 1.5% in C_{avg} (from 2.3% to 0.7%). These results confirm that the bottleneck features contain additional valuable information exploitable for SLI.

4.4.2. CNN architecture

The second architecture we focus on is that of convolutional neural networks. In contrast to Lozano-Diez et al. (2015), we also employ the bottleneck features (DNN-1l-48ph) in addition to FBCs.

The utilized CNN was composed of 2 convolutional layers and 3 fully connected layers (each with 1024 neurons). The inputs consists of 31 feature maps (i.e., the 0.3-second context window as before) at a size of 39×1. The first convolutional layer is comprised of 105 feature maps at a size of 39×1, followed by a 3:1 max-pooling layer. The second one has 157 feature maps, 13×1 in size. The rest of the hyper-parameters were set as stated in Section 4.4.1.

The results are covered in the second row of Table 3. As expected, the DNN-1l-48ph yields better results than FBCs by a large margin (by almost 4%). Interestingly, the difference in the performance of DNNs and CNNs is practically negligible; consequently, the more complex CNN architecture is not necessary. Note that different configurations of CNNs were explored in our previous study (Mateju et al., 2018).

4.4.3. RNN architectures

In the next step, we explore the recurrent neural network architecture. We focus on both LSTM RNNs (e.g., Gonzalez-Dominguez et al., 2014; Zazo et al., 2016) and GRU RNNs (e.g., Geng et al., 2016b). In addition to unidirectional RNNs, we have also studied the bidirectional ones (e.g., Fernando et al., 2017; however, we utilize a slightly different configuration here).

Each RNN consists of two recurrent layers (with 1024 neurons per layer) and two fully connected layers (each with 1024 neurons as well). As previously, we employ the same FBCs and DNN-1l-48ph BTNs as input features, with the same context window size of 0.3 seconds. The remaining hyper-parameters are set the same as in Section 4.4.1. Note that, for unidirectional models, the final decision for each utterance has been made by averaging only the last 10% of the frame probabilities, as suggested in Zazo et al. (2016), to exploit the learning capabilities of RNNs.

Table 3
Results of various NN-based classifiers for FBCs and DNN-1l-48ph bottleneck features (for the same data as in Table 2).

features	FBCs		DNN-1l-48ph BTNs	
NN architecture	ER [%]	C_{avg} [%]	ER [%]	C_{avg} [%]
DNN	4.8	2.6	1.2	0.7
CNN	4.9	2.7	1.3	0.7
LSTM RNN	3.0	1.7	1.4	0.7
GRU RNN	2.5	1.4	1.2	0.7
BLSTM RNN	3.0	1.6	1.5	0.8
BGRU RNN	2.4	1.3	1.2	0.6
TDNN	**2.0**	**1.1**	**1.0**	**0.5**
FSMN	2.2	1.2	1.2	0.7

The results are presented in rows 3–6 in Table 3. They indicate several key points. First, both the FBCs and bottleneck features outperform the baseline i-vector system (i.e., the first time for FBCs) with the DNN-1l-48ph BTNs being supreme. Additionally, the difference in performance between FBCs and DNN-1l-48ph BTNs becomes distinctly smaller. Recurrent neural networks can thus extract more information about the target language from the standard acoustic features. Second, GRU-based RNNs yield slightly better results than LSTMs (e.g., 1.4% vs. 1.2% ER). Last, the best results have so far been obtained by employing the BGRU RNN (see row 6). However, the improvement from feed-forward fully connected DNN to BGRU RNN is only 0.1% in C_{avg} at the cost of more complex architecture to train and evaluate (rows 1 and 6).

4.4.4. TDNN architecture

The next utilized architecture is represented by the time-delay neural networks. TDNNs using bottleneck features were successfully used for SLI in, e.g., Garcia-Romero and McCree (2016). However, our approach differs in the configuration. Specifically, our trained TDNN has been designed to match the fully connected feed-forward DNN. For this reason, it consists of five hidden layers, each with 1024 neurons. The input context of each layer required to compute output at one time step includes three preceding inputs, the current input, and three following inputs (from the preceding layer). This setting matches the input context window size of the feed-forward DNN (i.e., 0.3 s). As usual, the remaining hyper-parameters are unchanged, and FBCs and DNN-1l-48ph BTNs are utilized as input features.

The seventh row of Table 3 shows the results. It is obvious that the DNN-1l-48ph BTNs outperform the FBCs, although the difference has been reduced to a mere 1% in ER. Finally, in our setting, the TDNN yields the best results throughout the evaluation. It is evident that the long-term modeling capabilities of TDNNs (and RNNs) are exploitable in SLI.

4.4.5. FSMN architecture

The last topology used for classification is based on feed-forward sequential memory networks (see Section 3.2.2). To match the previous settings, it has 5 hidden layers, each with 1024 neurons, and the context is the same as for the TDNN-based classifier.

The results (see the last row of Table 3) are worse than for the similar TDNN-based architecture. On the other hand, we show in the next sections that FSMNs yield better results as BTN extractors.

4.5. Investigated bottleneck features

In the preceding section, we showed and confirmed the general opinion expressed in the literature; namely, the bottleneck features are more discriminating than the standard acoustic features (in this case, FBCs) for the task of SLI. This section thus focuses on several different bottleneck features extracted using various techniques and different data. Note that we already covered our bottleneck extractors in detail in Section 3. Therefore, this section is mainly focused on experimental evaluation. For all experiments, we employ a TDNN-based classifier (as described in Section 4.4.4) as it has yielded the best results so far (see Table 3).

4.5.1. DNN-based bottleneck features

We first compare our monolingual DNN-1l-48ph BTNs (used extensively in Section 4.4) with the multilingual features of the same architecture but trained using all 11 Slavic languages and four different phoneme sets (containing from 70 to 14 units).

The results are summarized in the first section of Table 4. They clearly show that the multilingual features are beneficial (improvement of at least 0.5% in ER) and that the multilingual bottleneck extractors are capable of capturing more of the language-specific information. Additionally, a fairly small phoneme set of 27 phonemes yields the best results as a compromise between merging too few and too many similar phonemes of different Slavic languages.

Table 4
Results in the off-line mode for TDNN-based classifier using various bottleneck features (for the same data as in Tables 2 and 3).

BTN features	ER [%]	C_{avg} [%]
DNN-1l-48ph	0.98	0.54
DNN-11l-70ph	0.47	0.26
DNN-11l-36ph	0.44	0.24
DNN-11l-27ph	0.35	0.19
DNN-11l-14ph	0.40	0.22
TDNN-11l-27ph	0.31	0.17
FSMN-11l-27ph	**0.20**	**0.11**
FSMN-11l-BS+data	**0.20**	**0.11**
FSMN-11l-BS+data+aug	**0.20**	**0.11**

4.5.2. Context modeling bottleneck features

In the next step, we first focus on the architecture of the extractors. We switch the architecture from DNN to TDNN and FSMN (as described in Section 3.2). At first, we use the training data labeled using the 27-phoneme set to make a direct comparison possible. The results can be seen in the fourth, sixth and seventh row of Table 4. It is evident that the more complex TDNN and FSMN architectures do further improve the results and that FSMN-based BTNs outperform the TDNN-based ones.

In the follow-up step, FSMN with a block soft-max output layer is employed, allowing us to not merge the phoneme sets of individual languages. Additional data has also been added as well as a significant amount of augmented data (see Section 3.2.2 for details). The results can be seen in the last two rows of Table 4. They show that no further reduction in ER or C_{avg} has been achieved. However, the benefits of these BTNs can be observed in the results for more difficult on-line task and the real-world data in Section 8.

4.6. Error analysis and confusion matrices

Figs. 1 and 2 show more detailed results in the form of confusion matrices for the baseline i-vector system and the best performing TDNN + FSMN-11l-27ph system, respectively. They show that most of the errors are the misclassification instances between closely related languages. They are mostly caused by a common phonetic inventory but also because the languages share common words in vocabulary and similar phonotactics, as a wider context is used for SLI.

For the baseline system, the most errors have been detected between Belorussian and Ukrainian (green highlights in Fig. 1). These two languages are more similar to each other than to Russian (while all the three belong to the East Slavic branch) as they phonetically differ only in a few phonemes, and they have similar vocabularies. A comparable case can be made between

	CZ	SK	PL	RU	SI	UA	RS	MK	HR	BY	BG
CZ	472	**10**	**4**	3	1	0	0	0	**9**	1	0
SK	**5**	483	**5**	0	0	0	3	1	0	3	0
PL	**9**	**5**	478	**5**	0	0	0	0	1	1	1
RU	5	3	**9**	470	1	2	0	0	**8**	1	1
SI	0	1	0	0	479	4	**11**	0	2	1	2
UA	1	0	1	1	0	481	0	0	3	**9**	4
RS	0	3	0	0	**10**	2	477	1	**5**	0	2
MK	0	1	0	0	0	2	4	489	1	0	3
HR	**8**	1	2	**5**	1	0	**6**	0	476	0	1
BY	0	1	1	0	4	**15**	0	2	2	471	4
BG	0	0	0	0	3	0	2	2	0	0	493

Fig. 1. A confusion matrix produced by the baseline i-vector system.

	CZ	SK	PL	RU	SI	UA	RS	MK	HR	BY	BG
CZ	497	3	0	0	0	0	0	0	0	0	0
SK	1	499	0	0	0	0	0	0	0	0	0
PL	0	0	500	0	0	0	0	0	0	0	0
RU	0	0	0	500	0	0	0	0	0	0	0
SI	0	0	0	0	500	0	0	0	0	0	0
UA	0	0	0	0	0	499	0	0	0	1	0
RS	0	1	0	0	0	0	497	0	2	0	0
MK	0	0	0	0	0	0	3	497	0	0	0
HR	0	0	0	0	0	0	0	0	500	0	0
BY	0	0	0	0	0	0	0	0	0	500	0
BG	0	0	0	0	0	0	0	0	0	0	500

Fig. 2. A confusion matrix produced by the best performing off-line system (TDNN + FSMN-11l-27ph BTNs).

Croatian, Serbian, and Slovene (red cluster) since the first two have the same phonetic inventory, and Slovene differs only in a few phonemes. They also share very similar vocabularies. The blue cluster highlights all West Slavic languages which get confused with each other. However, their phonetic inventories are not so close, and the source of confusion may be their similar vocabularies and phonotactics.

In some other cases, the confusions are harder to explain and may lay in the nature of the recordings (e.g., acoustic conditions, source of recordings, speaker characteristics) rather than in the mutual closeness of the languages. These errors are highlighted in yellow. Confusions between Polish and Russian are a good example. In this case, Russian is much closer to other East Slavic languages. Note that most of these errors (almost all) are eliminated by using the best TDNN + FSMN-11l-27ph system as depicted in Fig. 2.

5. Proposed approach for on-line SLI

The proposed scheme closely follows an on-line speech activity detection approach we proposed in Mateju et al. (2017) and a method we developed for on-line speaker change point detection in Mateju et al. (2019). It represents an extension of the off-line processing pipeline and utilizes a language classier consuming the input stream of BTN features, whose output is smoothed by a WFST-based decoder (see Fig. 3).

Note that this concept allows us to utilize some findings from the off-line evaluation. In other words, we assume that the optimal NN architecture for off-line mode (i.e., the TDNN-based classifier) should yield very good results also for a limited amount of input speech frames within a streamed environment. We performed several experiments confirming this assumption, and we made the decision not to repeat all off-line experiments in the on-line mode. This allows us to keep the on-line evaluation in the following sections much more concentrated just on the specific aspects of this mode.

The decoding scheme consists of two transducers. The first one models the input signal (see Fig. 4). The second one represents the transduction model. Its basic example for two languages is depicted in Fig. 5. The number of states of the transduction model corresponds to the number of languages (state 1 and 2 for the first and second language, respectively) plus one state 0 is used as the initial one. The transition rules between states are denoted by symbols L1 and L2, each representing one language being recognized by the DNN classifier under the hood. In other words, the classifier in the example distinguishes, for each input frame, between two languages L1 and L2 (i.e., it assigns to either of these languages a probability on the frame level). Note that in every step of the decoding process, when one of the transition rules (L1 or L2) is applied (as the transducer moves to another state or persists in the same state as in the previous frame), the same symbol (L1 or L2) is also generated on the output of the decoder.

The transducer is weighted so that the transitions between states 1 and 2 are penalized by factors P1 and P2, respectively, whose values have to be determined on the development set. Note that, in practice, both these penalties (or even all penalty fac-

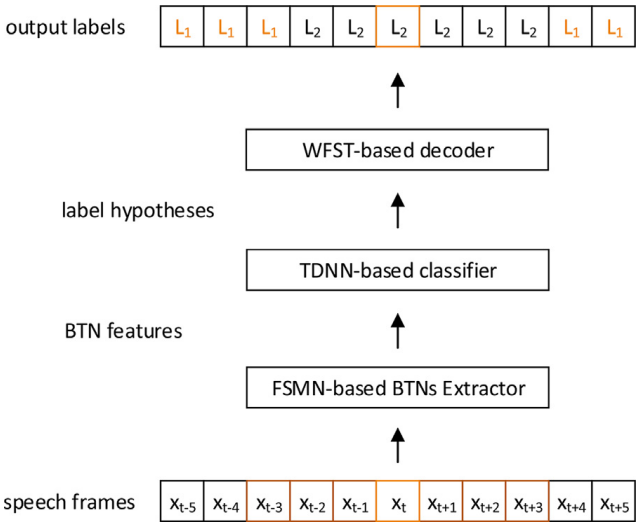


Fig. 3. The proposed scheme for on-line spoken language identification.

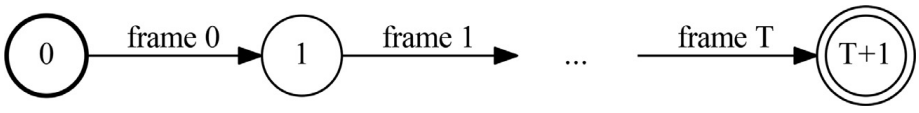


Fig. 4. The transducer representing the input signal.

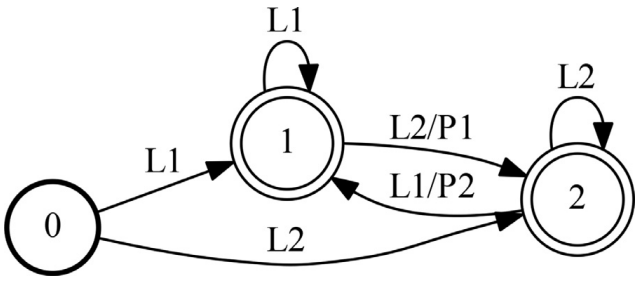


Fig. 5. The transducer (acceptor) representing the basic transduction model for two languages.

tors in the case of more than two languages) may be set to the same value. The sensitivity of the obtained results with respect to the penalty factor values is analyzed in detail in Section 8.2.

Given the two transducers described above, the decoding process is performed using on-the-fly composition of the transduction and the input model of an unknown size. That method is applicable here because the input is considered to be a linear-topology, unweighted, epsilon-free acceptor. After each composition step, the shortest-path (considering the tropical semi-ring) determined in the resulting model is compared with all other alternative hypotheses. When a common path (i.e., with the same output label) is found among these hypotheses, the corresponding concatenated output labels are marked as the resulting fixed output. Since the remaining proportion of the best path is not known with certainty, it is denoted as a temporary output (i.e., it can be further refined).

5.1. Context transduction model

The transduction model described so far represents just the basic possibility of the decoding scheme. In Mateju et al. (2017), our final approach for speech activity detection (SAD) utilized a context-based transduction model instead of the basic one depicted in Fig. 5. This improved the performance of SAD noticeably. For this reason, we also adopted this context-based transduction model for SLI.

The context-based transducer is shown in Fig. 6 for two languages, but it can easily be extended to several more. In this case, each of the two languages is represented as a sequence of three states, where the context is modeled by transitions. As previously, the penalty weights P1 and P2 are only defined for the transitions between the two languages, i.e., a) from the end state of L1 (*stop_L*1) to the start state of L2 (*start_L*2), and b) from the end state of L2 (*stop_L*2) to the start state of L1 (*start_L*1).

Due to the fact that each language is now covered by three states, the training data must be relabeled accordingly. Whenever a transition from L1 to L2 occurs, n frames before and n frames after the actual change point are labeled as *stop_L*1 and *start_L*2, respectively. By analogy, if a change from L2 to L1 occurs, the transition frames are labeled *end_L*2 and *start_L*1. Except for these transition frames, all remaining frames are labeled either L1 or L2 according to the corresponding language. The value of n is explored in Section 8.3. Finally, each language has to be represented by three neurons (e.g., *start_L*1, L1, *stop_L*1) in the output layer of the neural network. The other hyper-parameters remain unchanged.

6. Evaluation metrics for on-line SLI

For evaluation within the real-time SLI scenario, precision (P), recall (R) and F-measure (F) are employed. All these measures can be expressed given the alignment between the detected and reference language change points (Rasanen et al., 2009). The measure frame error rate (FER) determines the percentage of input speech frames marked with a wrong language label.

The measure named latency (L), has been employed to monitor the performance from a real-time processing point of view. It represents an average time between the input frame occurrence and the moment the decoder outputs its language label. Note that the presented latency values include the latency of the BTN extractor (around 500 ms), NN used for classification (around 187.5 ms for TDNN) and the decoding process itself.

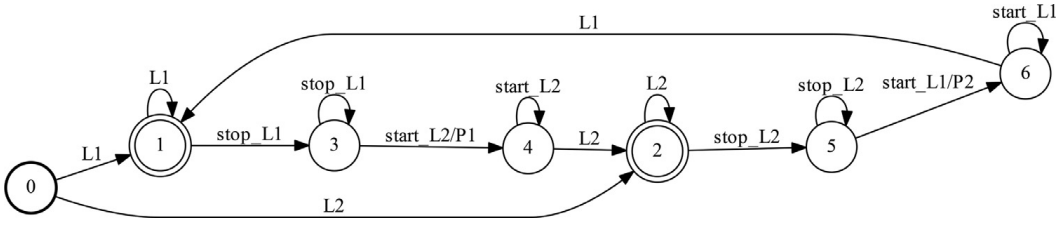


Fig. 6. The transducer (acceptor) representing the context transduction model for two languages.

7. Evaluation for artificial multilingual data

At first, the proposed on-line SLI scheme is evaluated on artificially prepared rather than real-world data, which has to be annotated manually. Moreover, the evaluation is also simpler as the prepared data does not contain any cross-talks, non-speech segments, etc. It is also true that if the system does not work well on this simpler type of the data, it will not work on real-world data either.

7.1. Data used for training

For training purposes, the same Slavic dataset is used as for the off-line scenario. To simulate transitions between two different languages, the training utterances were shuffled and joined in a random order. Given the fact that the average length of an utterance is a few seconds, and there are 220 hours of training data available, a sufficient number of transitions between all possible language combinations is ensured. This also includes joined segments in which the language remains unchanged, but the speaker or source differs.

7.2. Data used for testing

The evaluation data is treated in the same way as the training data. That means that all 5500 utterances (500 per language) of the Slavic dataset (evaluation subset) were shuffled and joined in a random order. To ease manipulation of the data, the joined recording was split into 100 evaluation files, each with a 5-min duration (i.e., each contains approximately 60 original utterances).

Note that the final data chunks also contain concatenated segments of the same language, where the speaker/source is different. This setup is intentional to make sure the network is learning to discriminate between languages and not between speakers or different acoustic channels.

7.3. Obtained results

Given the aforementioned datasets for training and testing, the experiments are conducted using the TDNN classifier, which yields the best results in the off-line scenario. Its build-in latency is 5 (number of layers) x 3 (half of the context) x 10 (frame-shift), i.e., 150 ms for DNN- and TDNN-based BTNs and 5 x 3 x 12.5 = 187.5 ms for FSMN-based BTNs.

At first, the TDNN classifier utilized FBCs and its output was not smoothed. The obtained results in the first row of Table 5 show that this approach does not work at all. The reached precision is just 0.8%, recall 100% and FER 90.1%, which means that the system simple marks (wrongly) most frames as containing a language different from the previous frame. For a classifier employing DNN-11l-27ph BTNs, the result without any smoothing in terms of P, R and F-measure are similarly bad (see the second row in Table 5). The only exception is FER of 29.3%, which shows much higher accuracy of BTNs against FBCs on the frame level – the difference is much more evident than for the off-line scenario, where identification was performed over a whole recording (utterance).

In the next experiment, the output from the TDNN-based classifier with DNN-11l-27ph BTNs was smoothed using a moving average (MA) method with different contexts. The results can be seen in Table 6. They clearly show the necessity of smoothing as all metrics have significantly improved (see the second row of Table 5 for comparison). It is also evident that the context window

Table 5
Performance of the TDNN-based classifier without smoothing and with FBCs and DNN-11l-27ph BTNs.

features	P [%]	R [%]	F [%]	FER [%]	L [s]
FBCs	0.8	100	1.5	90.1	0.4
DNN-11l-27ph BTNs	1.2	100	2.3	29.3	0.9

Table 6
Performance of the TDNN-based classifier with DNN-11l-27ph BTNs and smoothing based on moving average.

Context [s]	P [%]	R [%]	F [%]	FER [%]	L [s]
1	19.7	54.0	28.9	8.8	**1.2**
1.5	36.0	67.3	46.9	7.0	1.7
2	50.5	76.3	60.8	**6.6**	2.2
2.5	60.1	81.8	69.3	6.7	2.7
3	**67.1**	**84.3**	**74.8**	7.3	3.2

Table 7
Performance of the TDNN-based classifier with smoothing using the WFST decoder for various features.

Features	P [%]	R [%]	F [%]	FER [%]	L [s]
FBCs	43.6	43.7	43.6	88.4	4.5
DNN-11l-27ph BTNs	97.2	97.1	97.1	3.8	**2.7**
TDNN-11l-27ph BTNs	96.9	95.8	96.4	3.8	3.0
FSMN-11l-27ph BTNs	97.1	96.3	96.7	**3.1**	2.9
FSMN-11l-BS+data BTNs	97.8	97.3	97.5	3.2	2.9
FSMN-11l-BS+data+aug BTNs	**98.1**	**97.5**	**97.8**	3.2	2.9

size for MA smoothing influences the results noticeably. The best-performing variant of MA from the FER point of view utilized a 2-s context (i.e., for each frame, 1 second prior and 1 second after the current frame were used to compute the average).

In the last experiment for the artificial data-set, the WFST-based decoder was used. Its advantage is that it represents a general smoothing concept and allows us to model various sorts of smoothing approaches by merely choosing a corresponding transduction model and respective semi-ring (see also Section 5 for the description of two transduction models employed within this work). In this experiment, the basic transduction model with one global penalty factor set to 140 was employed (the influence of this factor for all important metrics including WER is discussed and analyzed in detail for real-world data in Section 8.2).

The obtained results in the second row of Table 7 clearly show that this approach has outperformed MA smoothing by a large margin. For example, the absolute difference in FER against MA with 2-second context (see the third row of Table 6) is around 3%.

The last three rows of Table 7 show that BTNs extracted by using FSMNs yield the best results. These values, e.g., F-measure of 97.8%, are very high. However, the evaluation in the next section shows that real-world bilingual broadcast shows are much more difficult for language identification.

From the latency point of view, it is evident that the better the features are (or the classifier is), the lower latency the system has. In other words, the decoder is able to output the resulting label earlier if the classifier operates at a higher accuracy level (the latency for FBCs is 4.5 s, while for the best FSMN-11l-BS+data+aug BTNs it is just 2.9 s.) Note that all latency values are measured by using processor Intel Core i7-3770K @ 3.50GHz.

8. Tuning and performance evaluation for bilingual real-world broadcast shows

The results presented so far have been obtained on artificially prepared mixtures of monolingual speech segments. In the next phase, the evaluation progressed towards real-world data. The development set consisted of ten recordings of a talk-show from a Slovak TV channel. Their total length was 3 hours and they contained 29,215 words. 12,288 of them belonged to a Slovak presenter of the show, the remaining 16,927 words to 10 different Czech guests of the shows. Note that all the persons in this dataset are native speakers and the presenter is fully proficient (this holds also for a test dataset used in Section 9).

A typical radio or TV stream contains long sections of music and many other non-speech events. Therefore, our broadcast transcription platform uses a speech/non-speech detector at the input (Mateju et al., 2017), which filters the non-speech segments so that only the speech segments are then further processed. This speech detector therefore influences the results of all other modules and the on-line language recognizer will also be deployed over this system. In order to eliminate the influence of the accuracy of this module and, at the same time, simulate its existence, the evaluation of language identification and speech transcription takes place over manually annotated speech segments of individual programs from the development set.

That means that the reference manual annotations of the development data include speech/non-speech and language change points as well as text transcriptions in both languages. At first, these annotations are used to calculate the reference WER: all speech segments belonging to one of the two languages are transcribed by the corresponding Czech or Slovak ASR system. Both these ASR systems utilize the FSMN-11l-BS+data+aug architecture (without the BTN layer) for acoustic modeling and n-gram language models. The Czech and Slovak lexicons contain 400k and 360k words, respectively. As a result of this process, the reference WER of 21.9% has been reached.

In addition to the already mentioned ten recordings of the Slovak talk-show, the evaluation was also performed on monolingual Czech and Slovak broadcast programs in terms of FER_{cz} and FER_{sk}, respectively. The reason is that the proposed scheme must be capable of continuous deployment on the input data stream and must not degrade the results for monolingual programs. The monolingual recordings used consist of news, debates and also the mentioned Slovak talk-show. Their total length is 7 h.

8.1. Training data for TDNN-based classifier

At first, the same setup as within the evaluation in Section 7.3 is used, i.e., the artificial dataset for all 11 Slavic languages are applied to the training of the TDNN classifier, WFST decoder, and different BTNs.

The obtained results (see the first part of Table 8) are much worse for all architectures than in the previous case. It means that, for real talk-shows, the transitions between languages are much more difficult to discriminate between than for mixtures prepared artificially. This namely holds for the DNN-based BTNs, which are outperformed in all measures and with a large difference by the TDNN-based as well as FSMN-based BTNs. For example, FER_{cz} is 86.8% for DNN-11l-27ph BTNs, while much lower value of

Table 8
Results [%] for various BTN extractors on the development set depending on the data used for training of the TDNN-based classifier. The reference WER is 21.9%.

Type of BTNs	P	R	F	FER	WER	FER_{cz}	FER_{sk}	L [s]
data: 11 lang., 220h artif.								
DNN-11l-27ph	15.7	16.4	16.0	34.8	72.5	86.8	28.1	3.4
TDNN-11l-27ph	46.3	26.7	33.9	13.6	39.3	33.1	18.0	4.4
FSMN-11l-27ph	33.5	27.8	30.4	16.0	39.6	28.7	17.5	4.1
FSMN-11l-BS+data	35.2	37.2	36.2	15.8	38.3	12.9	18.9	3.1
FSMN-11l-BS+data+aug	37.2	36.4	36.8	15.3	**37.2**	11.3	18.9	3.3
data: 2 lang., 40h artif.								
DNN-11l-27ph	18.9	31.6	23.7	29.5	67.5	66.3	5.4	2.1
TDNN-11l-27ph	29.2	45.8	35.7	14.5	44.9	33.0	4.4	2.1
FSMN-11l-27ph	30.7	42.8	35.7	15.6	40.2	33.7	8.4	2.6
FSMN-11l-BS+data	25.4	44.9	32.5	18.4	43.4	11.4	26.5	2.0
FSMN-11l-BS+data+aug	38.5	49.3	43.2	13.0	**35.7**	10.7	14.7	2.3
data: 2 lang., 40h artif. + 40h broadcast								
DNN-11l-27ph	51.7	56.9	54.2	8.2	31.6	4.4	6.5	2.3
TDNN-11l-27ph	72.4	45.7	56.0	7.8	30.8	1.4	2.1	3.0
FSMN-11l-27ph	64.4	46.3	53.8	7.3	30.1	3.0	1.5	3.0
FSMN-11l-BS+data	63.3	65.0	64.1	5.8	**27.7**	2.8	1.7	2.2
FSMN-11l-BS+data+aug	68.2	61.9	64.9	5.8	27.9	3.2	1.0	2.3
data: 2 lang., 40h artif. + 80h broadcast + 20 talk-shows and interviews								
DNN-11l-27ph	50.9	40.5	45.1	12.4	31.6	5.6	0.1	2.8
TDNN-11l-27ph	74.6	46.4	57.3	7.6	29.9	1.7	0.3	3.2
FSMN-11l-27ph	70.3	48.9	57.7	7.1	29.8	1.6	0.3	2.8
FSMN-11l-BS+data	69.1	62.1	65.4	5.5	27.1	0.8	0.6	2.2
FSMN-11l-BS+data+aug	74.4	62.2	67.8	5.0	**26.8**	0.9	0.2	2.2
data: 2 lang., 40h artif. + 80h brodcast + 20h talk-shows and interviews + cross-val.								
FSMN-11l-BS+data+aug	83.3	64.2	72.5	4.4	**26.3**	0.4	0.8	2.5

11.3% is yielded by the best FSMN-11l-BS+data+aug BTNs. Similarly, the measure most important from a practical point of view, the WER, is 72.5% for the first type of BTNs, but almost twice as low (37.2%) for the best FSMN-based BTNs. However, even the values for the best FSMN-based BTNs are still far from suitable.

Therefore, the set of languages is limited just to Czech and Slovak in the next experiment. The amount of the training data has accordingly been decreased from 220 to 40 h.

This step brought an overall improvement for all architectures (see the second part of Table 8). The results have gotten worse just in a few cases, e.g., WER for FSMN-11l-BS+data BTNs has been decreased from 38.3% to 43.4%. The most important improvement is reached for the latency, which has been decreased below 3 seconds for all systems as not all 11 languages had to be evaluated during the decoding process.

To further improve the results, the training dataset is extended by incorporating 40 h of monolingual Czech and Slovak broadcast programs. This additional data does not include any recording of the target talk-show (i.e., the system is not trained for the voice of the Slovak presenter).

The obtained results, shown in the third part of Table 8, provide high and important improvement in all measures. For example, for the FSMN-11l-BS+data+aug system as compared with the previous experiment, the WER value has been decreased from 35.7% to 27.9%, and the FER_{cz} and FER_{sk} from 10.7% to 3.2% and from 14.7% to 1.0%, respectively. The best values are reached by using FSMN-11l-BS+data BTNs, which somewhat outperform the FSMN-11l-BS+data+aug extractor.

A natural idea now is to further extend the training dataset. For this purpose, an additional 60 h of monolingual programs are used, out of which 20 h are the recordings of the target talk-shows and interviews.

As a result, a slight additional improvement in almost all measures can be observed for the TDNN-based and FSMN-based architectures in the fourth part of Table 8. This namely holds for the best FSMN-11l-BS+data+aug BTNs, where WER has been decreased from 27.9% to 26.8%. Also FER_{cz} and FER_{sk} have both been reduced to very low values – smaller than 1%. On the contrary, the DNN-based extractor fails to learn anything new from the additional data and the results do not improve or even get worse. For example, the WER value has remained the same at 31.6%, the F-measure value has been decreased from 54.2% to 45.1%, etc. The only exception is FER_{sk}, which has coincidentally been decreased to 0.1%. As in most previous experiments, the TDNN-based BTNs has been outperformed slightly by the FSMN-based extractor.

Finally, the recordings of the programs from the development set are used for the training within cross-validation. The reason is that all the evaluated systems have so far seen no data with real transitions between languages. They could learn the transitions just from the boundaries of randomly joined monolingual segments in the artificial dataset. Note that this experiment has been performed just for the best FSMN-11l-BS+data+aug features.

The results are summarized in the last row of Table 8. In this case, most measures have somewhat been improved, e.g., the WER value has been decreased from 26.8% to 26.3%. This value is absolutely, by 4.4%, higher than the reference WER of 21.9% based on the manually determined language change points. FER_{cz} as well as FER_{sk} has remained under 1% and the final average latency is 2.5 s.

8.2. The influence of the penalty factor

As mentioned in Section 5, the penalty factor P_f of the decoder has been 140 in all experiments performed so far. Table 9 and Fig. 7 present the results in terms of WER, FER and F-measure for P_f ranging from 1 to 400. The obtained values show that:

- from all measures evaluated, the F-measure changes in a most dynamic way with a changing value of P_f;
- the lowest value of WER is reached for P_f set to 120;
- the stable region is, for all measures, between 100 and 250;
- the differences in WER values are negligible in this area, i.e., the value of P_f should be somewhere in the stable area, but it need not necessarily hit the optimum value.

Finally, Fig. 7 also documents the correspondence between all three measures used. Namely, very strong correlation between FER and WER can be observed.

8.3. The context transduction model for decoding

In the follow-up experiment, we have replaced the basic transduction model in our decoding scheme with the context-based one as described in Section 5.1, similar to speech/non-speech detection in Mateju et al. (2017). We set the number of the transition frames to 100 (i.e., 50 frames around the actual change point), 50, and 30 (this setting corresponds to the input feature context size). For this experiment, we only trained TDNN using the FSMN-11l-BS+data+aug bottlenecks.

The results are shown in Table 10. They show that the context transduction model does not yield any additional improvements in the SLI performance. The transduction models using 30 and 50 transitional frames perform in a way rather similar to the basic model, and the results yielded by using 100 transitional frames are even worse. One thing to point out here is that the transduction model has improved the performance of some of the recordings but has notably worsened other ones. Lastly, as expected, the latency is increased with additional frames.

Table 9
The dependence of F-measure, WER and FER on decoder's penalty factor P_f.

P_f	1	10	50	100	120	140	160	180	200	250	300	400
F	5.1	16.4	50.4	67.6	**68.0**	67.8	67.1	66.8	66.0	63.0	59.9	57.0
FER	20.8	15.5	7.6	5.4	**5.0**	**5.0**	5.1	5.2	5.2	5.5	6.1	6.5
WER	53.3	46.0	29.5	26.9	**26.7**	26.8	26.9	27.0	27.1	27.6	28.5	29.1

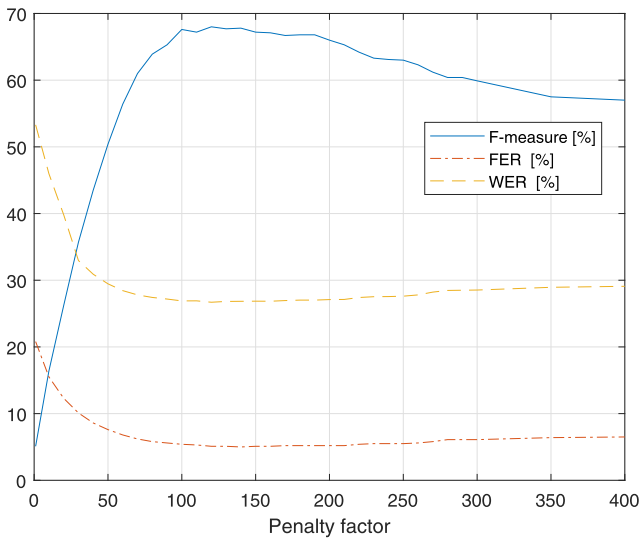


Fig. 7. The dependence of F-measure, WER and FER on decoder's penalty factor P_f.

Table 10
Results [%] of the context transduction model on the development set.

Transition frames	P	R	F	FER	WER	FER_{cz}	FER_{sk}	L [s]
none	**74.4**	**62.2**	**67.8**	**5.0**	**26.8**	0.9	0.2	**2.2**
30	65.1	59.2	62.0	6.8	29.0	1.6	0.3	2.3
50	75.9	56.0	64.5	5.2	27.0	**0.2**	0.8	2.7
100	37.8	24.4	29.7	19.7	49.2	7.7	**0.1**	3.7

Table 11
Detailed results [%] of the proposed on-line SLI approach on the test set.

Recording	1	2	3	4	5	6	7	8	9	10	Average
ref. WER	20.4	21.2	20.9	15.0	14.7	17.3	15.5	**10.9**	14.1	22.1	18.1
WER	24.6	23.6	23.0	18.9	21.8	23.5	16.8	**12.8**	18.7	26.0	21.0
FER	4.7	2.0	2.6	3.3	5.1	5.6	**1.5**	2.1	3.7	4.0	3.4
F	69.3	**93.3**	63.7	76.8	66.0	63.9	84.9	80.0	68.9	55.1	71.5
L [s]	2.3	**2.1**	3.3	2.2	3.0	2.8	2.3	2.2	2.4	3.3	2.6

9. Detailed evaluation on an independent test set

Finally, given the results on the development set, the performance of the proposed approach is evaluated on a test set. While the development stage of the work used Slovak broadcast shows with Slovak hosts and Czech guests, the chosen test data represents an opposite situation. It contains ten recordings of interviews from a Czech TV channel. Their total length is 3.25 h and they contain 30,089 words. 7617 of them belong to three different Czech presenters of the program, and the remaining 22,472 words to ten different Slovak guests. The reference WER value on this dataset is 18.1%.

The evaluation is performed using the FSMN-11l-BS+data+aug architecture for BTN extraction, the WFST decoder employing the basic transduction model with P_f set to 120, and the TDNN-based classifier trained on the data corresponding to the last row of Table 8 (i.e., cross validation has been performed). In addition, the development bilingual data has also been used.

The results obtained for individual recordings as well as their average values are summarized in Table 11. They confirm the results yielded on the development set and show that the scheme proposed for the on-line SLI has the following features:

1. it operates with a low average built-in latency around 2.6 s (the worst average delay observed is 3.3 s for the third and tenth recording, and the lowest 2.1 s for the second recording);
2. it yields an average WER value by a mere few percent higher than the best possible reference value (the average WER is 21.0%, the reference one 18.1%, and their difference amounts to 2.9%).

These results also show that the variance of an increase in the WER value against the reference is 3.2%; the worst reached increase is 7.1% for the fifth recording, and the smallest increase is a mere 1.3% for the eighth recording.

It should also be noted that from the latency point of view, our ASR system uses FSMN-based acoustic models with a build-in latency of 550 ms. With this delay, it is possible to get the best current ASR hypothesis. However, the delay for obtaining a fixed hypothesis that can no longer be changed is higher and depends on the type of input data. For broadcast streams, it is usually around 1.5 s. This is still a lower number than the latency of the proposed on-line SLI method. In the case of a combination of both systems on streamed data where low latency is required, it is possible to perform ASR in parallel for both (or all) languages and to switch the already existing output from individual ASR streams according to the results of the SLI module. The resulting latency is then given just by the SLI module, and it is not further increased by the ASR module.

10. Conclusions

In this work, a new approach suitable for on-line SLI is proposed in a series of consecutive experiments starting from an off-line mode through tests on artificial data to on-line processing of real broadcast programs.

From evaluation in the first off-line scenario for 11 Slavic languages, the following conclusions can be drawn:

1. Bottleneck features are beneficial for all investigated DNN architectures, and yield the lowest error rates.
2. The best results are obtained by using the TDNN architecture for the classification.
3. The best bottleneck features are those derived from the FSMN topology and trained as multilingual on all available data with various augmentations (the FSMN-11l-BS+data+aug extractor).
4. The evaluation set consists of recordings no longer than 5 s so that the resulting configuration may even be utilized for short recordings.

The more detailed analysis of these results in the form of confusion matrices further shows that, according to our assumptions, the worst results have in most cases been reached for groups of languages that are closely related to each other and belong

to the same branches of Slavic languages (i.e., they are also difficult to be distinguished from each other by humans). These most challenging groups are Belorussian and Ukrainian (East branch), Czech, Slovak and Polish (West branch) and Serbian, Croatian and Slovene (from the South branch). The best system with the TDNN classifier and the FSMN-11l-BS+data+aug BTNs is able to reduce mistakes for groups of languages with low (as well as high) baseline error rates (i.e., throughout the whole confusion matrix).

The on-line scenario has been proved to be much more challenging. First, the evaluation on the artificial data with the best off-line system shows that, for frame-by-frame language identification, it is necessary to smooth the output from the classifier. For this purpose, the WFST-based decoder has been introduced.

After that, the proposed SLI scheme is evaluated and tuned up on the bilingual development set consisting of recordings of real Slovak TV talk-shows that contain Czech and Slovak utterances. The goal here is to find the data suitable for training, select the proper transduction model for the decoder, and show the influence of the decoder's penalty factor.

Finally, given all the findings obtained so far, the best system with optimally set parameters has been evaluated on the test data consisting of interview recordings (again containing a mixture of Czech and Slovak utterances, but this time from a Czech TV station). The results obtained prove that the proposed SLI method is capable of operating in the frame-by-frame mode with latency below three seconds and with WER just a few percent above the reference value established by Czech and Slovak ASR systems manually switched by a human supervisor.

These results allow the deployment of the proposed approach within our production environment to improve on-line transcription of bilingual programs containing mixtures of utterances in Czech and Slovak languages, which allows for running reliable TV/R alert service mentioned in the introduction part. This should result in a gradual increase in the amount of available data containing real transitions between these two languages, which can subsequently be corrected and used for further training in order to increase the accuracy of the proposed method.

A concluding remark: The presented research has been done on spoken data representing Slavic languages, Czech and Slovak in particular, because we have had large amounts of this data needed for the experiments from our previous projects. For Czech and Slovak, moreover, we were able to prepare very detailed annotations that were necessary for the evaluation and error analysis. Recently, we have been using the experience and the proposed approach in a project that includes also Scandinavian languages where a similar phenomenon of language mixing frequently occurs.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Technology Agency of the Czech Republic (Project No. TH03010018 and Project No. TO01000027), and by the Student Grant Competition of the Technical University of Liberec under project No. SGS-2019-3017.

References

web. NIST Language Recognition Evaluations2020. http://nist.gov/itl/iad/mig/lre.cfm, Online (accessed: 2020-05-20).

LRE. 2015. The 2015 NIST language recognition evaluation plan (LRE15).

LRE, 2017. NIST 2017 language recognition evaluation plan.

Abdullah, B. M., Avgustinova, T., Möbius, B., Klakow, D., 2020. Cross-domain adaptation of spoken language identification for related languages: the curious case of slavic languages. 2008.00545.

Cai, W., Cai, D., Huang, S., Li, M., 2019. Utterance-level end-to-end language identification using attention-based CNN-BLSTM. 2019 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12–17, 2019. IEEE, pp. 5991–5995. https://doi.org/10.1109/ICASSP.2019.8682386.

Cai, W., Cai, Z., Liu, W., Wang, X., Li, M., 2018. Insights in-to-end learning scheme for language identification. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15–20, 2018. IEEE, pp. 5209–5213. https://doi.org/10.1109/ICASSP.2018.8462026.

Cai, W., Cai, Z., Zhang, X., Wang, X., Li, M., 2018. A novel learnable dictionary encoding layer for end-to-end language identification. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15–20, 2018. IEEE, pp. 5189–5193. https://doi.org/10.1109/ICASSP.2018.8462025.

Cai, W., Chen, J., Li, M., 2018. Exploring the encoding layer and loss function in end-to-end speaker and language recognition system. Odyssey 2018: The Speaker and Language Recognition Workshop, Les Sables d'Olonne, France, June 26–29, 2018. ISCA, pp. 74–81. https://doi.org/10.21437/Odyssey.2018-11.

Caseiro, D., Trancoso, I., 1998. Spoken language identification using the speechdat corpus. 5th International Conference on Spoken Language Processing (ICSLP), Sydney, Australia, November 30, - December 4, 1998. ISCA, pp. 1–4.

Dahl, G.E., Yu, D., Deng, L., Acero, A., 2012. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Trans. Audio Speech Lang. Process. 20 (1), 30–42. https://doi.org/10.1109/TASL.2011.2134090.

Dehak, N., Torres-Carrasquillo, P.A., Reynolds, D.A., Dehak, R., 2011. Language recognition via i-vectors and dimensionality reduction. INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27–31, 2011. ISCA, pp. 857–860.

D'Haro, L.F., de Cordoba, R., Palacios, C.S., Echeverry, J.D., 2014. Extended phone log-likelihood ratio features and acoustic-based i-vectors for language recognition. 2014 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4–9, 2014. IEEE, pp. 5342–5346. https://doi.org/10.1109/ICASSP.2014.6854623.

Fer, R., Matejka, P., Grezl, F., Plchot, O., Cernocky, J., 2015. Multilingual bottleneck features for language recognition. INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6–10, 2015. ISCA, pp. 389–393.

Fer, R., Matejka, P., Grezl, F., Plchot, O., Vesely, K., Cernocky, J.H., 2017. Multilingually trained bottleneck features in spoken language recognition. Comput. Speech Lang. 46, 252–267. https://doi.org/10.1016/j.csl.2017.06.008.

Fernando, S., Sethu, V., Ambikairajah, E., Epps, J., 2017. Bidirectional modelling for short duration language identification. INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20–24, 2017. ISCA, pp. 2809–2813. https://doi.org/10.21437/Interspeech.2017-286.

Ferrer, L., Lei, Y., McLaren, M., Scheffer, N., 2016. Study of senone-based deep neural network approaches for spoken language recognition. IEEE/ACM Trans. Audio Speech Lang. Process. 24 (1), 105–116. https://doi.org/10.1109/TASLP.2015.2496226.

Garcia-Romero, D., McCree, A., 2016. Stacked long-term TDNN for spoken language recognition. INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8–12, 2016. ISCA, pp. 3226–3230. https://doi.org/10.21437/Interspeech.2016-1334.

Gauvain, J., Messaoudi, A., Schwenk, H., 2004. Language recognition using phone latices. INTERSPEECH 2004 - ICSLP, 8th International Conference on Spoken Language Processing, Jeju Island, Korea, October 4–8, 2004. ISCA, pp. 1283–1286.

Gelly, G., Gauvain, J., 2017. Spoken language identification using LSTM-based angular proximity. INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20–24, 2017. ISCA, pp. 2566–2570. https://doi.org/10.21437/Interspeech.2017-1334.

Gelly, G., Gauvain, J., Le, V.B., Messaoudi, A., 2016. A divide-and-conquer approach for language identification based on recurrent neural networks. INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8–12, 2016. ISCA, pp. 3231–3235. https://doi.org/10.21437/Interspeech.2016-180.

Geng, W., Wang, W., Zhao, Y., Cai, X., Xu, B., 2016. End-to-end language identification using attention-based recurrent neural networks. INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8–12, 2016. ISCA, pp. 2944–2948. https://doi.org/10.21437/Interspeech.2016-686.

Geng, W., Zhao, Y., Wang, W., Cai, X., Xu, B., 2016. Gating recurrent enhanced memory neural networks on language identification. INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8–12, 2016. ISCA, pp. 3280–3284. https://doi.org/10.21437/Interspeech.2016-684.

Gonzalez, D.M., Plchot, O., Burget, L., Glembek, O., Matejka, P., 2011. Language recognition in ivectors space. INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27–31, 2011. ISCA, pp. 861–864.

Gonzalez-Dominguez, J., Lopez-Moreno, I., Sak, H., Gonzalez-Rodriguez, J., Moreno, P.J., 2014. Automatic language identification using long short-term memory recurrent neural networks. INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14–18, 2014. ISCA, pp. 2155–2159.

Griol, D., Molina, J.M., Sanchis, A., Callejas, Z., 2020. A data-driven approach to spoken dialog segmentation. Neurocomputing 391, 292–304. https://doi.org/10.1016/j.neucom.2019.02.072.

Jin, M., Song, Y., McLoughlin, I.V., Dai, L., 2018. Lid-senones and their statistics for language identification. IEEE/ACM Trans. Audio Speech Lang. Process. 26 (1), 171–183. https://doi.org/10.1109/TASLP.2017.2766023.

Li, H., Ma, B., Lee, C., 2007. A vector space modeling approach to spoken language identification. IEEE Trans. Audio Speech Lang. Process. 15 (1), 271–284. https://doi.org/10.1109/TASL.2006.876860.

Li, H., Ma, B., Lee, K., 2013. Spoken language recognition: from fundamentals to practice. Proc. IEEE 101 (5), 1136–1159. https://doi.org/10.1109/JPROC.2012.2237151.

Lim, D.C.Y., Lane, I.R., Waibel, A., 2010. Real-time spoken language identification and recognition for speech-to-speech translation. 2010 International Workshop on Spoken Language Translation, IWSLT 2010, Paris, France, December 2–3, 2010. ISCA, pp. 307–312.

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., Alsaadi, F.E., 2017. A survey of deep neural network architectures and their applications. Neurocomputing 234, 11–26. https://doi.org/10.1016/j.neucom.2016.12.038.

Lopez, J.A.V., Brummer, N., Dehak, N., 2018. End-to-end versus embedding neural networks for language recognition in mismatched conditions. Odyssey 2018: The Speaker and Language Recognition Workshop, Les Sables d'Olonne, France, June 26–29, 2018. ISCA, pp. 112–119. https://doi.org/10.21437/Odyssey.2018-16.

Lopez-Moreno, I., Gonzalez-Dominguez, J., Plchot, O., Martinez, D., Gonzalez-Rodriguez, J., Moreno, P.J., 2014. Automatic language identification using deep neural networks. 2014 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4–9, 2014. IEEE, pp. 5337–5341. https://doi.org/10.1109/ICASSP.2014.6854622.

Lozano-Diez, A., Plchot, O., Matejka, P., Gonzalez-Rodriguez, J., 2018. DNN based embeddings for language recognition. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15–20, 2018. IEEE, pp. 5184–5188. https://doi.org/10.1109/ICASSP.2018.8462403.

Lozano-Diez, A., Zazo-Candil, R., Gonzalez-Dominguez, J., Toledano, D.T., Gonzalez-Rodriguez, J., 2015. An end-to-end approach to language identification in short utterances using convolutional neural networks. INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6–10, 2015. ISCA, pp. 403–407.

Malek, J., Zdansky, J., 2019. On practical aspects of multi-condition training based on augmentation for reverberation-/noise-robust speech recognition. Text, Speech, and Dialogue – 22nd International Conference, TSD 2019, Ljubljana, Slovenia, September 11–13, 2019. Springer, pp. 251–263. https://doi.org/10.1007/978-3-030-27947-9_21.

Malek, J., Zdansky, J., Cerva, P., 2018. Robust recognition of conversational telephone speech via multi-condition training and data augmentation. Text, Speech, and Dialogue - 21st International Conference, TSD 2018, Brno, Czech Republic, September 11–14, 2018. Springer, pp. 324–333. https://doi.org/10.1007/978-3-030-00794-2_35.

Masumura, R., Asami, T., Masataki, H., Aono, Y., 2017. Parallel phonetically aware DNNS and LSTM-RNNS for frame-by-frame discriminative modeling of spoken language identification. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5–9, 2017. IEEE, pp. 5260–5264. https://doi.org/10.1109/ICASSP.2017.7953160.

Mateju, L., Cerva, P., Zdansky, J., 2019. An approach to online speaker change point detection using DNNs and WFSTs. INTERSPEECH 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, September 15–19, 2019. ISCA, pp. 649–653. https://doi.org/10.21437/Interspeech.2019-1407.

Mateju, L., Cerva, P., Zdansky, J., Malek, J., 2017. Speech activity detection in online broadcast transcription using deep neural networks and weighted finite state transducers. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5–9, 2017. IEEE, pp. 5460–5464. https://doi.org/10.1109/ICASSP.2017.7953200.

Mateju, L., Cerva, P., Zdansky, J., Safarik, R., 2018. Using deep neural networks for identification of slavic languages from acoustic signal. INTERSPEECH 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, September 2–6, 2018. ISCA, pp. 1803–1807. https://doi.org/10.21437/Interspeech.2018-1165.

McLaren, M., Ferrer, L., Lawson, A., 2016. Exploring the role of phonetic bottleneck features for speaker and language recognition. 2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20–25, 2016. IEEE, pp. 5575–5579. https://doi.org/10.1109/ICASSP.2016.7472744.

Miao, X., McLoughlin, I., Yan, Y., 2019. A new time-frequency attention mechanism for TDNN and cnn-lstm-tdnn, with application to language identification. INTERSPEECH 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, September 15–19, 2019. ISCA, pp. 4080–4084. https://doi.org/10.21437/Interspeech.2019-1256.

Mingote, V., Castan, D., McLaren, M., Nandwana, M.K., Gimenez, A.O., Lleida, E., Miguel, A., 2019. Language recognition using triplet neural networks. INTERSPEECH 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, September 15–19, 2019. ISCA, pp. 4025–4029. https://doi.org/10.21437/Interspeech.2019-2437.

Nouza, J., Safarik, R., Cerva, P., 2016. ASR for south slavic languages developed in almost automated way. INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8–12, 2016. ISCA, pp. 3868–3872. https://doi.org/10.21437/Interspeech.2016-747.

Okamoto, T., Hiroe, A., Kawai, H., 2017. Reducing latency for language identification based on large-vocabulary continuous speech recognition. Acoust. Sci. Technol. 38 (1), 38–41. https://doi.org/10.1250/ast.38.38.

Padi, B., Mohan, A., Ganapathy, S., 2019. Attention based hybrid i-vector BLSTM model for language recognition. INTERSPEECH 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, September 15–19, 2019. ISCA, pp. 1263–1267. https://doi.org/10.21437/Interspeech.2019-2371.

Padi, B., Mohan, A., Ganapathy, S., 2019. End-to-end language recognition using attention based hierarchical gated recurrent unit models. 2019 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12–17, 2019. IEEE, pp. 5966–5970. https://doi.org/10.1109/ICASSP.2019.8683895.

Pesan, J., Burget, L., Cernocky, J., 2016. Sequence summarizing neural networks for spoken language recognition. INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8–12, 2016. ISCA, pp. 3285–3288. https://doi.org/10.21437/Interspeech.2016-764.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., Vesely, K., 2011. The Kaldi speech recognition toolkit. 2011 IEEE Workshop on Automatic Speech Recognition & Understanding, ASRU 2011, Waikoloa, HI, USA, December 11–15, 2011. IEEE, pp. 1–4.

Rasanen, O.J., Laine, U.K., Altosaar, T., 2009. An improved speech segmentation quality measure: the r-value. INTERSPEECH 2009, 10th Annual Conference of the International Speech Communication Association, Brighton, United Kingdom, September 6–10, 2009. ISCA, pp. 1851–1854.

Richardson, F., Reynolds, D.A., Dehak, N., 2015. Deep neural network approaches to speaker and language recognition. IEEE Signal Process. Lett. 22 (10), 1671–1675. https://doi.org/10.1109/LSP.2015.2420092.

Richardson, F., Reynolds, D.A., Dehak, N., 2015. A unified deep neural network for speaker and language recognition. INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6–10, 2015. ISCA, pp. 1146–1150.

Singer, E., Torres-Carrasquillo, P.A., Reynolds, D.A., McCree, A., Richardson, F., Dehak, N., Sturim, D.E., 2012. The MITLL NIST LRE 2011 language recognition system. Odyssey 2012: The Speaker and Language Recognition Workshop, Singapore, June 25–28, 2012. ISCA, pp. 209–215.

Siniscalchi, S.M., Svendsen, T., Lee, C., 2014. An artificial neural network approach to automatic speech processing. Neurocomputing 140, 326–338. https://doi.org/10.1016/j.neucom.2014.03.005.

Snyder, D., Garcia-Romero, D., McCree, A., Sell, G., Povey, D., Khudanpur, S., 2018. Spoken language recognition using x-vectors. Odyssey 2018: The Speaker and Language Recognition Workshop, Les Sables d'Olonne, France, June 26–29, 2018. ISCA, pp. 105–111. https://doi.org/10.21437/Odyssey.2018-15.

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., Khudanpur, S., 2018. X-vectors: robust DNN embeddings for speaker recognition. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15–20, 2018. IEEE, pp. 5329–5333. https://doi.org/10.1109/ICASSP.2018.8461375.

Song, Y., Hong, X., Jiang, B., Cui, R., McLoughlin, I.V., Dai, L., 2015. Deep bottleneck network based i-vector representation for language identification. INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6–10, 2015. ISCA, pp. 398–402.

V., M.K., Achanta, S., R., L.H., Gangashetty, S.V., Vuppala, A.K., 2016. An investigation of deep neural network architectures for language recognition in indian languages. INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8–12, 2016. ISCA, pp. 2930–2933. https://doi.org/10.21437/Interspeech.2016-910.

Wan, L., Sridhar, P., Yu, Y., Wang, Q., Lopez-Moreno, I., 2019. Tuplemax loss for language identification. 2019 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12–17, 2019. IEEE, pp. 5976–5980. https://doi.org/10.1109/ICASSP.2019.8683313.

Zazo, R., Lozano-Diez, A., Gonzalez-Rodriguez, J., 2016. Evaluation of an LSTM-RNN system in different NIST language recognition frameworks. Odyssey 2016: The Speaker and Language Recognition Workshop, Bilbao, Spain, June 21–24, 2016. ISCA, pp. 231–236. https://doi.org/10.21437/Odyssey.2016-33.

Zhang, S., Liu, C., Jiang, H., Wei, S., Dai, L., Hu, Y., 2015. Feedforward sequential memory networks: A new structure to learn long-term dependency. CoRR.abs/1512.08301

Zissman, M.A., 1996. Comparison of four approaches to automatic language identification of telephone speech. IEEE Trans. Audio Speech Process. 4 (1), 31. https://doi.org/10.1109/TSA.1996.481450.

6 Conclusions

This thesis deals with methods allowing ASR systems to adapt to selected real-world deployment conditions. These conditions affect the functionality or recognition accuracy of almost every ASR system and are related to the variability of input data. The considered sources of this variability comprise in particular

- specific voice characteristics of individual speakers,
- occurrence of non-speech events in the audio signal,
- input utterances in multiple languages.

These three mentioned factors are not chosen at random. On the contrary, each of them represents an important issue that must actually be solved when many ASR systems are being deployed.

The previous statement is based on the experience I have gained over the last decade as a member of the speech-processing group at TUL. In this period, our team has created the core of several ASR systems that are deployed in practice and used on a daily basis by many people. This for example holds for voice control tools for motor-disabled persons or in the software for voice dictation, which is mainly used by medical doctors, lawyers or judges.

6.1 An example of a practically deployed complex ASR system

Currently, the most important and complex example of a deployed ASR application is the platform for 24/7 monitoring of TV/R broadcasts. It is used by many media monitoring companies in various European countries because the transcription process runs here in parallel for about a hundred TV/R channels in 15 languages.

The platform utilizes several of the methods that have been proposed within the articles included in this thesis, e.g., for frame-wise SAD or language adaptation. Other components, such as the LID module, are being integrated within the current TAČR project DeepSpot, of which I am a co-investigator. The live-running demo of a multilingual radio monitoring application (see also Fig. 6.1), which shows ASR capabilities of this platform, is available on the web[1].

[1] https://tul-speechlab.gitlab.io/

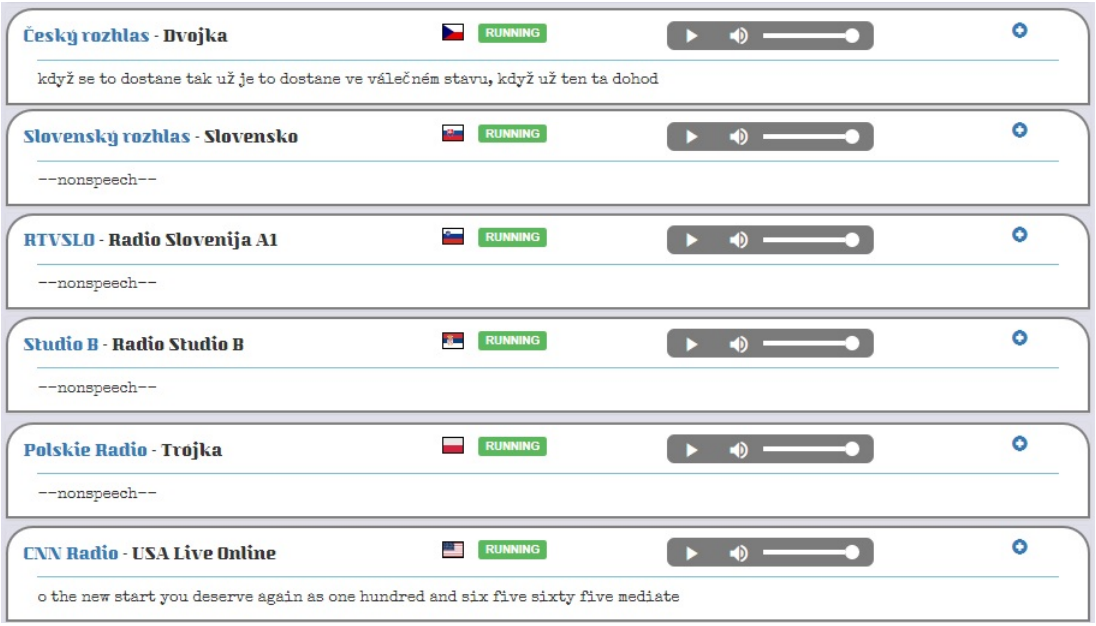


Figure 6.1: A screenshot of a demo application for multilingual radio monitoring.

6.2 Impact on education activities

My efforts in recent years has also been to transfer the issues addressed within our research to teaching. A few years ago I came up with the idea to create a new specialization of Intelligent systems within the existing bachelor's study program of Information technology and I coordinated the preparation of its curriculum.

This specialization has been launched at our faculty in 2020 and has two main directions. The first is theory in the field of signal processing and machine learning, which is covered in several interrelated subjects including Signals and information, Machine learning, Neural network applications, Introduction to image processing and Text mining methods. I myself teach the course of Machine learning and contribute to the teaching of Signals and information with several lectures. The second direction is represented by software technologies, that allow students to utilize the acquired theoretical knowledge in practice. It is represented by the following three subjects: Database for BigData, Technologies for BigData and Cloud technologies. Here, I am involved as their guarantor.

This specialization also forms the basis for the follow-up branch of Intelligent systems within the master's study program of Information technology. This branch contains subjects more specialized in individual areas of machine learning, such as Computer speech processing, Introduction to computational linguistics, etc. The best graduates of this branch are encouraged to enroll in the doctoral study at the faculty. Some of them have joined our research team where they have been working on the tasks related to this thesis. For example, the topic of my former PhD student Lukáš Matějů, who successfully defended his dissertation in 2020, was speech activity and speaker change point detection for on-line streams. The results of his research

have therefore also been applied in the development of the above-mentioned ASR systems and he currently teaches some of the subjects within the specialization of Intelligent systems. My recent PhD student focuses on methods for extraction of x-vectors and utilization of these embeddings for frame-wise speaker diarization. In this way, my experience gained during the 15-year research activities is transferred to the younger generations of students.

References

[1] P. Cerva, V. Volna, and L. Weingartova. "Dealing with Newly Emerging OOVs in Broadcast Programs by Daily Updates of the Lexicon and Language Model." In: *SPECOM*. Vol. 12335. Lecture Notes in Computer Science. Springer, 2020, pp. 97–107.

[2] J. Chaloupka, J. Nouza, P. Cerva, and J. Malek. "Downdating Lexicon and Language Model for Automatic Transcription of Czech Historical Spoken Documents." In: *TSD*. Vol. 8082. Lecture Notes in Computer Science. Springer, 2013, pp. 201–208.

[3] D. Wang, X. Wang, and S. Lv. "An Overview of End-to-End Automatic Speech Recognition." In: *Symmetry* 11.8 (2019). ISSN: 2073-8994.

[4] A. Graves and N. Jaitly. "Towards End-to-End Speech Recognition with Recurrent Neural Networks." In: *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*. ICML'14. Beijing, China: JMLR.org, 2014, pp. II–1764–II–1772.

[5] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, J. Chen, J. Chen, Z. Chen, M. Chrzanowski, A. Coates, G. Diamos, K. Ding, N. Du, E. Elsen, J. Engel, W. Fang, L. Fan, C. Fougner, L. Gao, C. Gong, A. Hannun, T. Han, L. V. Johannes, B. Jiang, C. Ju, B. Jun, P. LeGresley, L. Lin, J. Liu, Y. Liu, W. Li, X. Li, D. Ma, S. Narang, A. Ng, S. Ozair, Y. Peng, R. Prenger, S. Qian, Z. Quan, J. Raiman, V. Rao, S. Satheesh, D. Seetapun, S. Sengupta, K. Srinet, A. Sriram, H. Tang, L. Tang, C. Wang, J. Wang, K. Wang, Y. Wang, Z. Wang, Z. Wang, S. Wu, L. Wei, B. Xiao, W. Xie, Y. Xie, D. Yogatama, B. Yuan, J. Zhan, and Z. Zhu. "Deep Speech 2: End-to-End Speech Recognition in English and Mandarin." In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML'16. New York, NY, USA: JMLR.org, 2016, pp. 173–182.

[6] G. E. Dahl, D. Yu, L. Deng, and A. Acero. "Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition." In: *IEEE Transactions on Audio, Speech, and Language Processing* 20.1 (2012), pp. 30–42.

[7] H. Li, B. Ma, and K. Lee. "Spoken Language Recognition: From Fundamentals to Practice." In: *Proceedings of the IEEE* 101.5 (2013), pp. 1136–1159.

[8] M. Gales. “Maximum Likelihood Linear Transformations for HMM-Based Speech Recognition.” In: *COMPUTER SPEECH AND LANGUAGE* 12 (1998), pp. 75–98.

[9] M. L. Seltzer, D. Yu, and Y. Wang. “An investigation of deep neural networks for noise robust speech recognition.” In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2013, pp. 7398–7402.

[10] K. Kinoshita, M. Delcroix, T. Yoshioka, T. Nakatani, E. Habets, R. Haeb-Umbach, V. Leutnant, A. Sehr, W. Kellermann, R. Maas, S. Gannot, and B. Raj. “The reverb challenge: A common evaluation framework for dereverberation and recognition of reverberant speech.” In: *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. 2013, pp. 1–4.

[11] J. Malek, J. Zdansky, and P. Cerva. “Robust Recognition of Conversational Telephone Speech via Multi-condition Training and Data Augmentation.” In: *TSD*. Vol. 11107. Lecture Notes in Computer Science. Springer, 2018, pp. 324–333.

[12] T. Schultz. “GlobalPhone: A Multilingual Speech and Text Database Developed at Karlsruhe University.” In: *Proceedings of the ICSLP*. 2002, pp. 345–348.

[13] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury. “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups.” In: *IEEE Signal Processing Magazine* 29.6 (2012). cited By 5768, pp. 82–97.

[14] Y. Miao, H. Zhang, and F. Metze. “Towards speaker adaptive training of deep neural network acoustic models.” In: cited By 67. 2014, pp. 2189–2193.

[15] S. Parthasarathi, B. Hoffmeister, S. Matsoukas, A. Mandal, N. Strom, and S. Garimella. “FMLLR based feature-space speaker adaptation of DNN acoustic models.” In: vol. 2015-January. cited By 32. 2015, pp. 3630–3634.

[16] P. Cerva, J. Silovsky, J. Zdansky, J. Nouza, and J. Malek. “Real-Time Lecture Transcription using ASR for Czech Hearing Impaired or Deaf Students.” In: *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012*. ISCA, 2012, pp. 763–766.

[17] P. Cerva, K. Palecek, J. Silovsky, and J. Nouza. “Using Unsupervised Feature-Based Speaker Adaptation for Improved Transcription of Spoken Archives.” In: *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011*. ISCA, 2011, pp. 2565–2568.

[18] L. Welling, H. Ney, S. Kanthak, and L. F. I. Vi. “Speaker Adaptive Modeling by Vocal Tract Normalization.” In: *IEEE Trans. on Speech and Audio Processing* 10 (2002), pp. 415–426.

[19] S. Molau, S. Kanthak, and H. Ney. “Efficient Vocal Tract Normalization in Automatic Speech Recognition.” In: *In Proc. of the ESSV'00*. 2000, pp. 209–216.

[20] J. Silovsky, P. Cerva, J. Zdansky, and J. Nouza. “Study on Integration of Speaker Diarization with Speaker Adaptive Speech Recognition for Broadcast Transcription.” In: *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012*. ISCA, 2012, pp. 478–481.

[21] J. Silovsky and J. Prazak. “Speaker diarization of broadcast streams using two-stage clustering based on i-vectors and cosine distance scoring.” In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2012, pp. 4193–4196.

[22] P. Cerva, J. Silovsky, J. Zdansky, J. Nouza, and L. Seps. “Speaker-adaptive speech recognition using speaker diarization for improved transcription of large spoken archives.” In: *Speech Communication* 55.10 (2013), pp. 1033–1046.

[23] L. Mateju, P. Cerva, J. Zdansky, and J. Malek. “Speech Activity Detection in online broadcast transcription using Deep Neural Networks and Weighted Finite State Transducers.” In: *ICASSP*. IEEE, 2017, pp. 5460–5464.

[24] D. Dean, S. Sridharan, R. Vogt, and M. Mason. “The QUT-NOISE-TIMIT corpus for the evaluation of voice activity detection algorithms.” In: *INTERSPEECH*. ISCA, 2010, pp. 3110–3113.

[25] L. Mateju, F. Kynych, P. Cerva, J. Zdansky, and J. Malek. “Using X-vectors for Speech Activity Detection in Broadcast Streams.” In: *INTERSPEECH 2021, Annual Conference of the International Speech Communication Association, Brno, Czech Republic 2021*. ISCA, 2021, Accepted to.

[26] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur. “X-Vectors: Robust DNN Embeddings for Speaker Recognition.” In: *ICASSP 2018, Calgary, Canada*. 2018, pp. 5329–5333.

[27] S. Zhang, C. Liu, H. Jiang, S. Wei, L. Dai, and Y. Hu. “Feedforward Sequential Memory Networks: A New Structure to Learn Long-term Dependency.” In: *CoRR* abs/1512.08301 (2015).

[28] J. S. Chung, A. Nagrani, and A. Zisserman. “VoxCeleb2: Deep Speaker Recognition.” In: *Interspeech 2018, Hyderabad, India*. 2018, pp. 1086–1090.

[29] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. “Librispeech: An ASR corpus based on public domain audio books.” In: *ICASSP 2015, South Brisbane, Australia*. 2015, pp. 5206–5210.

[30] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer. “An analysis of environment, microphone and data simulation mismatches in robust speech recognition.” In: *Computer Speech & Language* 46 (2017), pp. 535–557.

[31] J. Malek, J. Zdansky, and P. Cerva. “Robust Automatic Recognition of Speech with background music.” In: *ICASSP*. IEEE, 2017, pp. 5210–5214.

[32] J. Malek, J. Zdansky, and P. Cerva. “Robust Recognition of Speech with Background Music in Acoustically Under-Resourced Scenarios.” In: *ICASSP*. IEEE, 2018, pp. 5624–5628.

[33] T. N. Sainath, B. Kingsbury, G. Saon, H. Soltau, A.-r. Mohamed, G. Dahl, and B. Ramabhadran. “Deep Convolutional Neural Networks for Large-scale Speech Tasks.” In: *Neural Networks* 64 (2015). Special Issue on “Deep Learning of Representations”, pp. 39–48. ISSN: 0893-6080.

[34] T. Gao, J. Du, L.-R. Dai, and C.-H. Lee. “Joint training of front-end and back-end deep neural networks for robust speech recognition.” In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015, pp. 4375–4379.

[35] J. Nouza, R. Safarik, and P. Cerva. “ASR for South Slavic Languages Developed in Almost Automated Way.” In: *INTERSPEECH*. ISCA, 2016, pp. 3868–3872.

[36] J. Nouza and R. Safarik. “Parliament Archives Used for Automatic Training of Multi-lingual Automatic Speech Recognition Systems.” In: *TSD*. Vol. 10415. Lecture Notes in Computer Science. Springer, 2017, pp. 174–182.

[37] J. Nouza, P. Cerva, and M. Kucharova. “Cost-efficient development of acoustic models for speech recognition of related languages.” In: *Radioengineering* 22.3 (2013), pp. 866–873.

[38] R. Safarik and L. Mateju. “Impact of phonetic annotation precision on automatic speech recognition systems.” In: *2016 39th International Conference on Telecommunications and Signal Processing (TSP)*. 2016, pp. 311–314.

[39] J. Nouza, P. Cerva, and R. Safarik. “Cross-Lingual Adaptation of Broadcast Transcription System to Polish Language Using Public Data Sources.” In: *Human Language Technology. Challenges for Computer Science and Linguistics - 7th Language and Technology Conference, LTC 2015, Poznań, Poland, November 27-29, 2015, Revised Selected Papers*. Ed. by Z. Vetulani, J. Mariani, and M. Kubis. Vol. 10930. Lecture Notes in Computer Science. Springer, 2015, pp. 31–41.

[40] S. Scanzio, P. Laface, L. Fissore, R. Gemello, and F. Mana. “On the use of a multilingual neural network front-end.” In: *INTERSPEECH*. 2008.

[41] K. Veselý, M. Karafiát, F. Grézl, M. Janda, and E. Egorova. “The language-independent bottleneck features.” In: *2012 IEEE Spoken Language Technology Workshop (SLT)*. 2012, pp. 336–341.

[42] L. Mateju, P. Cerva, J. Zdansky, and R. Safarik. “Using Deep Neural Networks for Identification of Slavic Languages from Acoustic Signal.” In: *INTERSPEECH*. ISCA, 2018, pp. 1803–1807.

[43] R. Fer, P. Matejka, F. Grezl, O. Plchot, K. Vesely, and J. H. Cernocky. “Multilingually trained bottleneck features in spoken language recognition.” In: *Compututer Speech & Language* 46 (2017), pp. 252–267.

[44] P. Cerva, L. Mateju, F. Kynych, J. Zdansky, and J. Nouza. “Identification of Scandinavian Languages from Speech Using Bottleneck Features and X-vectors.” In: *24th International Conference on Text, Speech, and Dialogue (TSD)*. Springer, 2021, Accepted to.

[45] J. Malek and J. Zdansky. “On Practical Aspects of Multi-condition Training Based on Augmentation for Reverberation-/Noise-Robust Speech Recognition.” In: *Text, Speech, and Dialogue*. Ed. by K. Ekštein. Cham: Springer International Publishing, 2019, pp. 251–263. ISBN: 978-3-030-27947-9.

[46] D. Garcia-Romero and A. McCree. “Stacked Long-Term TDNN for Spoken Language Recognition.” In: *Interspeech 2016, San Francisco, CA, USA*. 2016, pp. 3226–3230.

[47] P. Cerva, L. Mateju, J. Zdansky, R. Safarik, and J. Nouza. “Identification of related languages from spoken data: Moving from off-line to on-line scenario.” In: *Computer Speech and Language* 68 (2021).