BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

FIT

Advanced machine learning technologies for automatic speech recognition

Habilitation Thesis

Presented by: Ing. Petr Motlíček, Ph.D.
motlicek@fit.vutbr.cz

Submission: 30th March 2022

... to all brave friends from Ukraine.

Abstract

Automatic speech recognition (ASR) has become one of well-known machine learning technologies nowadays integrated in various consumer SW products. Although history of pursuing ASR is spread across many decades, recent improvements have enabled its potential integration in more critical applications such as air-traffic management (specifically automatic processing of spoken communication between air-traffic controllers and pilots). This thesis summarises our contributions towards solving generic ASR problems and demonstrates the capabilities of recent speech recognizers on typical applications with a particular focus on analysing air-traffic communication with the aim of supporting humans to be more efficient while reducing their workload.

Keywords

Automatic speech recognition, air-traffic communication, acoustic modeling, semi-supervised training, contextual boosting, language modeling, natural language understanding, named entity recognition.

Acknowledgements

First, I would like to thank my mentors in area of automatic speech recognition from Idiap (CH) and BUT (CZ) (alphabetically ordered): Hervé Bourlard, Honza Černocký and Hynek Heřmanský.

Second, I would like to thank all my co-authors from Idiap (CH), BUT (CZ) and DLR (DE) (alphabetically ordered): Ajay Srinivasamurthy, Amrutha Prasad, David Imseng, Esaú Villatoro-Tello, Gwénolé Lecorvé, Hartmut Helmke, Iuliia Nigmatulina, Juan Zuluaga-Gomez, Lukáš Burget, Matthias Kleinert, Oliver Ohneiser, Seyyed Saeed Sarfjoo and Srikanth Madikeri.

Last but not least, I would like to thank Daira, Karlík and Andrejka for supporting me, as well as my parents Štefánia and Vlastimil.

Over the time, I have been supported by a number of funding sources for research and innovation, in particular, by European Commission FP5, FP6, FP7 and H2020, IARPA U. S., ICT and Innosuisse (CH) as well as by numerous industrial funds.

Contents

Acknowledgements III
Figures VI
1 Introduction 1
1.1 Note on the author's contribution 2
1.2 Focus and structure of the thesis 5
2 Domain transfer learning in ASR 6
2.1 Introduction 6
2.2 Acoustic modeling 8
2.2.1 Generative models 8
2.2.2 Hybrid models 10
2.3 Language modeling 13
2.4 Domain and language transfer learning for acoustic and language modeling 14
2.4.1 Cross-lingual acoustic modeling 15
2.4.2 Multi-lingual acoustic modeling 18
2.4.3 Domain adaptation and multi-lingual training of LMs 20
2.5 Sequence discriminative training for AM 21
2.5.1 Extension to multi-lingual acoustic modeling 22
2.5.2 Extension to multi-lingual acoustic modeling and language transfer 23
2.6 Semi-supervised training 24
2.6.1 Data selection 24
2.6.2 Semi-supervised training using LF-MMI 25
2.6.3 Incremental semi-supervised training 25
2.6.4 Semi-supervised training for language modeling 26
2.7 Self-supervised acoustic model training 26
2.8 Application of ASR in air-traffic management 27
3 Boosting contextual information in ASR 31
3.1 Introduction 31
3.2 Boosting of OOV words 32
3.3 Boosting by a prior from another modality 32
3.4 Context-based re-scoring of ASR output 34
3.4.1 Contextual data in ATC 35
3.4.2 Context-based rescoring 36
3.5 Two-stage boosting 38
3.5.1 List of call-signs 38
3.5.2 Lattice boosting 38
3.5.3 HCLG boosting 39
3.6 Application of contextual boosting in air-traffic management 39
4 Natural language understanding on automatically generated textual data 40

4.1 Joint ASR and NLP 40
4.2 Supervised and unsupervised language model adaptation 41
4.2.1 N-gram LM adaptation 42
4.3 Conversion of RNN based LM to WFST 43
4.4 Boosting of NER for air-traffic management 44
4.4.1 Re-ranking 46
4.4.2 BERT - as speaker role detector 47
References 49
5 Selected papers underlying this thesis 55
5.1 Paper 1: [Ims+13b] 57
5.2 Paper 2: [Ims+13a] 68
5.3 Paper 3: [Mad+20a] 75
5.4 Paper 4: [MMB21] 81
5.5 Paper 5: [Sri+17] 87
5.6 Paper 6: [BMM21] 93
5.7 Paper 7: [MVG10] 99
5.8 Paper 8: [Oua+17] 104
5.9 Paper 9: [Koc+21b] 110
5.10 Paper 10: [Lec+12] 116
5.11 Paper 11: [LM12] 121
5.12 Paper 12: [Nig+21] 126

Figures

1 The author's most significant scientific contributions – 25 publications. Black denotes an essential contribution, grey denotes an important contribution, white denotes minor or no contribution, and crosses denote non-applicability. Out of 25 papers, 12 are selected (see those highlighted in red in Table 1) and commented on in this thesis. These 12 papers are also attached to this thesis (see Section 5). 3
2 Illustration of a typical HMM-based ASR system (source: [Bou+11]). 7
3 HMM/GMM acoustic model. 8
4 SGMM acoustic model (source: [Ims+13b]). 9
5 Tandem acoustic model. 11
6 HMM/DNN acoustic model. 12
7 KL-HMM acoustic model. 12
8 Tandem - the emission probabilities of the HMM states are modeled with Gaussian mixtures and the MLP output is postprocessed (source: [Ims+13b]). . 16
9 KL-HMM - the emission probabilities are modeled with categorical distributions and the MLP output can directly be used (source: [Ims+13b]). 17
10 Cross-lingual adaptation in DNN (source: [Mot+15]). 18
11 Single-task and multi-task adaptation for acoustic modeling (source: [MMB21]). 20
12 LF-MMI training for multi-lingual acoustic modeling (source: [Mad+20a]). 23
13 Incremental approach for semi-supervised training. Model100 means acoustic model developed using 100 hours of untranscribed data, etc. 25
14 Overview of air-traffic management system incorporating automatic speech recognition. 28
15 Output of ASR system developed for air-traffic management. 29
16 Automatic transcription and annotation of ATC speech data with possible manual verification. 30
17 Overall EERs of spoken-term detection when additional prior information is exploited: (a) STD system without OOL module, (b) STD system with OOL module (source: [MVG10]). 34
18 Schematic view of an automatic speech recognition-based ATC system (source: [Oua+17]). 35
19 Expected landing sequences and trajectories for different aircraft approaching Prague airport. (source: [Oua+17]). 36
20 Command extractor and corrector - steps described in Section 3.4.2. 37
21 Use of context-free grammar to transduce ATC segments (text) to concepts. . . 37
22 General overview of the proposed multi-stage SDR architecture based on expanded lattice embeddings (source: [VIL+22]). 41
23 Overview of the RNNLM discretization scheme. 44

24 Call-sign and command tagging: Named entity recognizer to automatically tag the input word sequence for respective ATC classes. Visualisation of BERT-based model (Huggingface) fine-tuned on NER task. 46
25 Integrate the surveillance data (extracted from radar screen) into the ASR system. 47
26 Overview of an ATM application of virtual pilot: besides ASR, the module for speaker role identification and text-to-speech (to generate an automatic response back to pilot) are deployed. 48

1 Introduction

Automatic Speech Recognition (ASR) - a sequence to sequence problem – remains a distinct field of research in speech and language technology, and builds around various machine learning algorithms making it a relatively complex task compared to other areas.

This thesis overviews our contributions in the context of automatic speech recognition. More specifically, in this thesis we decided to divide our contributions into three areas:

- Domain transfer learning in ASR (Section 2),
- Boosting contextual information in ASR (Section 3),
- Natural language understanding on automatically generated textual data (Section 4).

The reasons for such a division are as follows: ASR has been among our main interests over the last two decades. The first most evident open issues in the area of ASR were related to its applicability for new domains (environments) or languages. Here, we particularly consider low-resource scenarios which assume availability of relatively low amounts of development data. While this scenario may not be of interest for well-resourced (e.g. viable) languages or environments, there are numerous applications where direct deployment of ASR would fail (especially due to very low performance). Overview of our contributions in this area is given in Section 2.

Further, many applications comprising ASR require very accurate performance, either in terms of overall word error-rates[1], or at least when recognising highly informative set of words for subsequent downstream processes. This problem is specifically addressed in Section 3 where we consider boosting contextual information in ASR.

The third area builds on our contributions combining ASR and natural language understanding. More specifically, we comment on very recent works where automatically generated textual outputs (from ASR) are used as direct input for subsequent downstream applications, such as information (or spoken document) retrieval, or named entity recognition, including boosting these technologies using apriori known information. Besides above, Section 4 also comments on our work related to language modeling in ASR (concretely building the language models using supervised and unsupervised data). We also present our approach on using powerful recurrent neural networks for a direct decoding in ASR.

This thesis focuses with a large care on Air-Traffic Management (ATM). Recognition and understanding of communication between air-traffic controllers and pilots in the field of ATM has not been addressed in a sophisticated manner, until recently. We have started to work in this field

[1] https://en.wikipedia.org/wiki/Word_error_rate

in 2016. One of the reasons for starting to apply ASR into the ATM domain was due to recent significant improvements in acoustic modeling, thus reaching acceptable ASR accuracies in new environments even with limited development data. ASR in ATM will be addressed in all 3 areas of this thesis. Our most significant papers related to ATM are highlighted **in bold** in Table 1.

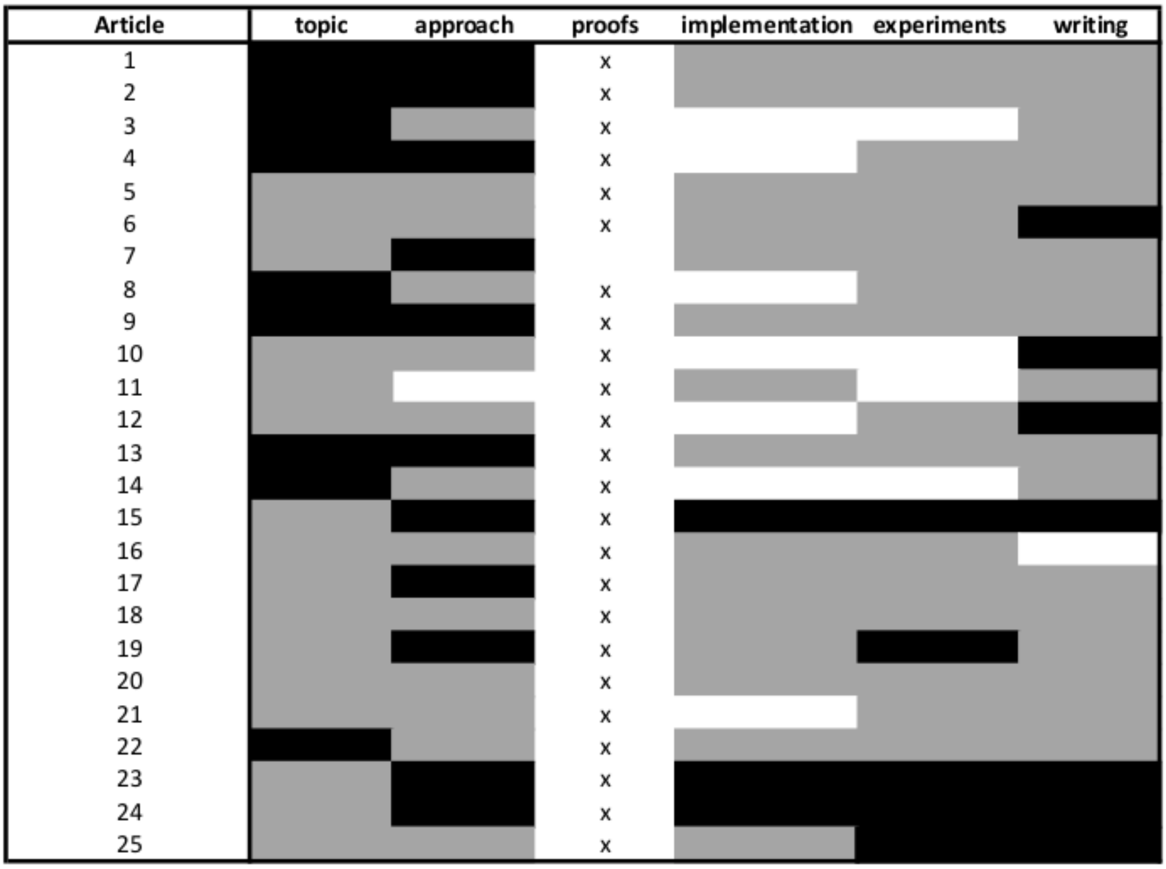
To address our contributions in those 3 aforementioned areas, we decided to select 12 papers to be introduced and commented on in this thesis. These 12 papers are attached in Section 5. Across all the following text, these 12 papers are also aligned with particular sections, and always highlighted in red. Besides these 12 papers, we also link some subsections with other potentially interesting papers - highlighted in blue.

Commentary of our works in this thesis does not include recognition/classification performances achieved in presented contributions. We rather give a high-level introduction to our works. Details are obviously presented in given papers.

1.1 Note on the author's contribution

We would like to emphasise that the 12 selected papers underlying this thesis (see Section 5 with the attachments) do not directly reflect authors' contribution. We therefore add Figure 1 which gives more details about the contribution of the author of this thesis to those papers. In fact, Figure 1 presents a super-set of 25 scientific publications (including those 12 papers). As there is no agreement on a metric allowing for a qualitative evaluation, the figure describes the author's contribution to commonly accepted parts of the process of creating a paper in computer science.

Following Table 1 presents a list of 25 scientific publications – ranked as the most significant according to the author. 6 papers in bold are related to the application domain of this thesis – “air-traffic management”. 12 selected papers (underlying this thesis) are highlighted in red.



Article	topic	approach	proofs	implementation	experiments	writing
1	■	■	x	▒	▒	▒
2	■	■	x	▒	▒	▒
3	■	▒	x	□	□	▒
4	■	■	x	□	▒	▒
5	▒	▒	x	▒	▒	▒
6	▒	▒	x	▒	▒	■
7	▒	■	□	▒	▒	▒
8	■	▒	x	□	▒	▒
9	■	■	x	▒	▒	▒
10	▒	▒	x	□	□	■
11	▒	□	x	▒	□	▒
12	▒	▒	x	□	▒	■
13	■	■	x	▒	▒	▒
14	■	▒	x	□	□	▒
15	▒	■	x	■	■	■
16	▒	▒	x	▒	▒	□
17	▒	■	x	▒	▒	▒
18	▒	▒	x	▒	▒	▒
19	▒	■	x	▒	■	▒
20	▒	▒	x	▒	▒	▒
21	▒	▒	x	□	▒	▒
22	■	▒	x	▒	▒	▒
23	▒	■	x	■	■	■
24	▒	■	x	■	■	■
25	▒	▒	x	▒	■	■

Figure 1 The author's most significant scientific contributions – 25 publications. Black denotes an essential contribution, grey denotes an important contribution, white denotes minor or no contribution, and crosses denote non-applicability. Out of 25 papers, 12 are selected (see those highlighted in red in Table 1) and commented on in this thesis. These 12 papers are also attached to this thesis (see Section 5).

Article	Title	Citation
1	**A two-step approach to leverage contextual data: speech recognition in air-traffic communications, Iuliia Nigmatulina et al., in: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) – to appear, 2022**	[Nig+22]
2	**Contextual Semi-Supervised Learning: An Approach To Leverage Air-Surveillance and Untranscribed ATC Data in ASR Systems, Juan Zuluaga-Gomez et al., in: Proceedings of Interspeech, 2021**	[Zul+21]
3	**Boosting of contextual information in ASR for air-traffic call-sign recognition, Martin Kocour et al., in: Proceedings of Interspeech, 2021**	[Koc+21b]
4	A COMPARISON OF METHODS FOR OOV-WORD RECOGNITION ON A NEW PUBLIC DATASET, Rudolf Braun et al., in: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, Ontario, Canada, 2021	[BMM21]
5	Multitask adaptation with Lattice-Free MMI for multi-genre speech recognition of low resource languages, Srikanth Madikeri et al., in: Proceedings of Interspeech, 2021	[MMB21]
6	Speech Activity Detection Based on Multilingual Speech Recognition System, Seyyed Saeed Sarfjoo et al., in: Proceedings of Interspeech, 2021	[SMM21]
7	Lattice-Free Maximum Mutual Information Training of Multilingual Speech Recognition System, Srikanth Madikeri et al., in: Proceedings of Interspeech, pages 4746–4750, ISCA, 2020	[Mad+20a]
8	INCREMENTAL SEMI-SUPERVISED LEARNING FOR MULTI-GENRE SPEECH RECOGNITION, Banriskhem Khonglah et al., in: Proceedings on IEEE International Conference on Acoustics, Speech and Signal Processing, 2020	[Kho+20]
9	**Automatic Speech Recognition Benchmark for Air-Traffic Communications, Juan Zuluaga-Gomez et al., in: Proceedings of Interspeech, pages 2297-2301, 2020**	[Zul+20b]
10	End-to-End Accented Speech Recognition, Thibault Viglino et al., in: Proceedings of Interspeech, ISCA, Graz, Austria, pages 2140-2144, 2019	[VMC19]
11	Abstract Text Summarization: A Low Resource Challenge, Shantipriya Parida et al., in: In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2019), HongKong, China, pages 5, Association for Computational Linguistics (ACL), 2019	[PM19]
12	Deep Neural Networks for Multiple Speaker Detection and Localization, Weipeng He et al., in: IEEE International Conference on Robotics and Automation (ICRA), Brisbane, AUSTRALIA, pages 74-79, 2018	[HMO18]
13	**Semi-supervised Learning with Semantic Knowledge Extraction for Improved Speech Recognition in Air Traffic Control, Ajay Srinivasamurthy et al., in: Proceedings of Interspeech, Stockholm, Sweden, pages 2406-2410, 2017**	[Sri+17]
14	**A Context-Aware Speech recognition and Understanding System for Air Traffic Control Domain, Youssef Oualil et al., in: Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop, Okinawa, Japan, 2017**	[Oua+17]
15	Exploiting foreign resources for DNN-based ASR, Petr Motlicek et al., in: EURASIP Journal on Audio, Speech, and Music Processing (2015:17), 2015	[Mot+15]
16	Multilingual Deep Neural Network based Acoustic Modeling For Rapid Language Adaptation, Ngoc Thang Vu et al., in: Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, pages 7639-7643, 2014	[Vu+14]
17	Using out-of-language data to improve an under-resourced speech recognizer, David Imseng et al., in: Speech Communication, pages 142-151, 2013	[Ims+13b]
18	Impact of deep MLP architecture on different acoustic modeling techniques for under-resourced speech recognition, David Imseng et al., in: IEEE Workshop on Automatic Speech Recognition and Understanding, pages 332-337, 2013	[Ims+13a]
19	Conversion of Recurrent Neural Network Language Models to Weighted Finite State Transducers for Automatic Speech Recognition, Gwénolé Lecorvé et al., in: Proceedings of Interspeech, Portland, Oregon, USA, pages 1666-1669, 2012	[LM12]
20	Comparing different acoustic modeling techniques for multilingual boosting, David Imseng et al., in: Proceedings of Interspeech, Portland, Oregon, 2012	[Ims+12]
21	Supervised and unsupervised Web-based language model domain adaptation, Gwénolé Lecorvé et al., in: Proceedings of Interspeech, Portland, Oregon, USA, 2012	[Lec+12]
22	The Kaldi Speech Recognition Toolkit, Daniel Povey et al., in: IEEE Workshop on Automatic Speech Recognition and Understanding, Hawaii, US, 2011	[Pov+11]
23	English Spoken Term Detection in Multilingual Recordings, Petr Motlicek et al., in: Proceedings of Interspeech, Makuhari, Japan, pages 206-209, 2010	[MVG10]
24	IMPROVING ACOUSTIC BASED KEYWORD SPOTTING USING LVCSR LATTICES, Petr Motlicek et al., in: Proceedings on IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Japan, pages 4413-4416, 2012	[MVS12]
25	Unsupervised Speech/Non-speech Detection for Automatic Speech Recognition in Meeting Rooms, Hari Krishna Maganti et al., in: Proceedings on IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2007	[MMG07]

Table 1 List of 25 scientific publications – ranked as the most significant according to the author. 6 papers in **bold** are related to the application domain of this thesis – “air-traffic management”. 12 papers in red are among those selected and commented in this thesis.

1.2 Focus and structure of the thesis

In the following sections, we will comment on and discuss in more detail some of the key ideas carried out by 12 selected papers underlying this thesis.

More particularly, Section 2 presents our achievements in the area of generic ASR with a particular attention to domain transfer learning. By domain transfer learning, we mean not only a new environment, but also possibly new language(s). 5 papers were selected to be commented on in this section. Both acoustic modeling as well as language modeling are targeted here although larger attention is given to R&D of acoustic models, i.e., dealing with the raw audio waveforms of human speech and predicting what phoneme each waveform corresponds to, typically at the character or sub-word level. Section 2 comments on our approaches in cross-lingual and multi-lingual modeling. It also presents our work in extending well-known sequence discriminative training for multi-lingual modeling and language transfer. Further, semi-supervised training for acoustic models is presented with our contributions in this area. We also introduce a self-supervised acoustic model training – a relatively new approach in ASR. This section eventually introduces the domain of air-traffic management and briefly overviews our work already done through various projects in applying ASR in this domain.

Section 3 is devoted to boosting of contextual information in ASR. Progress in contextual boosting – incorporating prior knowledge (e.g., from different modality) – for ASR has found many applications. First, boosting of out-of-vocabulary words for an artificially created test set (CommonVoice data) is presented. Then, we introduce an approach on spoken-term detection, allowing to detect a set of words in word recognition output, and boosting them by incorporating apriori information in form of textual data from corresponding PowerPoint presentations. Last three sub-sections (particularly Sections 3.4 to 3.6) are devoted to boosting contextual information in the area of air-traffic management. 4 papers were selected to be commented on in this section.

Section 4 comments on our past (including very recent) contributions in the domain of natural and spoken language understanding. First, this section reviews our work in the area of language modeling, particularly in supervised and unsupervised Language Model (LM) adaptation for ASR and conversion of conventional recurrent neural network based LMs to be used directly in the first pass of ASR decoders. Further, this section aims to highlight approaches which require the use of ASR to generate automatic transcripts used in subsequent downstream applications such as information retrieval and named entity recognition. 3 papers were selected to be commented on in this section.

The final Section 5 lists 12 selected papers underlying this thesis.

2 Domain transfer learning in ASR

This section summarises our R&D achievements in an area of automatic speech recognition and deployment of new algorithms for domain and language transfer learning . The presented work will mainly focus on acoustic modeling for ASR applications, nevertheless, language modeling will also be discussed here. Eventually, this section will also give a brief overview of aforementioned technologies for a particular domain targeted in this thesis – “air-traffic management” (Section 2.8).

Following 5 papers were selected among others to be summarised/commented and aligned with other works in this section:

(1) Using out-of-language data to improve an under-resourced speech recognizer, 2013 [Ims+13b],

(2) Impact of deep MLP architecture on different acoustic modeling techniques for under-resourced speech recognition, 2013 [Ims+13a],

(3) Lattice-Free Maximum Mutual Information Training of multi-lingual Speech Recognition System , 2020 [Mad+20a],

(4) Multitask adaptation with Lattice-Free MMI for multi-genre speech recognition of low resource languages, 2021 [MMB21],

(5) Semi-supervised Learning with Semantic Knowledge Extraction for Improved Speech Recognition in Air Traffic Control, 2017 [Sri+17].

2.1 Introduction

Current Automatic Speech Recognition (ASR) systems are based on statistical parametric methods. In the 1990s and in the first decade of the twentieth century, advances in ASR have been largely driven forward by the US government, specifically via the National Institute of Science and Technology (NIST)[2]. Besides participating on technology evaluations, one of the main contributions was related to collection of databases (e.g. [Kub+94], [PB92]). First works were naturally considering English language and a large progress was made to reach good performance on English related recognition tasks (i.e. read speech). Furthermore, these improvements were later translated directly to different languages, or domains (e.g. telephone speech, spontaneous speech, etc.), thanks largely to the robustness of statistical approaches to the different specificities of languages or domains.

Current conventional ASR systems are stochastic and their acoustic models (specifically hybrid types) still exploit Hidden Markov Models (HMM) in their framework. Most recent algorithms

[2] http://www.nist.gov/index.html

are built around deep neural learning as generative or discriminative modeling approaches. Figure 2 illustrates a typical HMM-based ASR system for the English language. The principal components of the HMM-based ASR system are as follows:

- Feature extraction: converting the speech sequence into a stream of feature observations/vectors. Feature extraction is typically considered a language-independent process. Among typical features, Mel-Frequency Cepstral Coefficients (MFCCs) [DM80] are still often used, composed of static coefficients and their approximate first order and second order time derivatives.
- Acoustic Model (AM): it models the relation between the speech feature vector(s) and units of spoken form (sound units, such as phones or directly graphemes). Development and various versions of AM will be particularly considered in this section with a focus on domain and language transfer.
- Lexicon: although in case of very recent end-to-end ASR approaches it is often unnecessary, lexicon (or dictionary) still plays an important role to integrate lexical constraints on top of spoken unit level representation yielding a unit representation that is typically common to both spoken form and written form such as, word or morpheme.
- Language Model (LM): LM represents syntactical/grammatical constraints of the spoken language. Often, it is modeled using statistical models such as n-grams, or more recently using deep neural network architectures (e.g., recurrent neural networks).

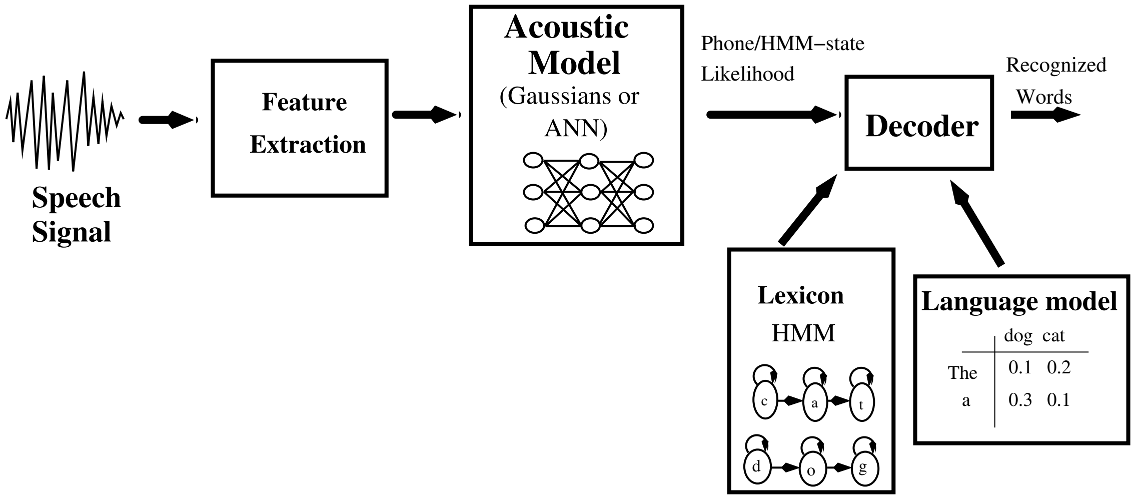


Figure 2 Illustration of a typical HMM-based ASR system (source: [Bou+11]).

2.2 Acoustic modeling

This section will review and discuss approaches applied for the development of the acoustic models for ASR. Specifically this section will describe (i) technologies to build both generative and discriminative acoustic models, (ii) approaches allowing to perform language and/or domain transfer learning (i.e., adaptation), (iii) approaches aimed to significantly increase robustness of the ASR models by applying sequence-discriminative training, and (iv) training procedures which allow us to employ both transcribed as well as untranscribed speech corpora for supervised and semi-supervised training.

2.2.1 Generative models

HMM/GMMs

Past acoustic models also applied in many production ASR systems were largely built around Gaussian Mixture Models (GMMs) (i.e., generative models) to model the speech feature observations in HMM/GMM architectures [Rab89] (see Figure 3). Specifically, this type of AM aimed to represent distributions of (usually tied) Hidden Markov Model (HMM) states using a relatively large number of parameters completely defining a GMM. This approach was also considered in the recent past as state-of-the-art in acoustic modeling especially for Large Vocabulary Continuous Speech Recognition (LVCSR). The main advantage of the HMM/GMM compared to other acoustic modeling techniques (see later those employing neural networks) was its feasibility for parallel training (i.e., it can easily accommodate large amounts of training data which is usually available for well-resourced languages) and possibility to combine standard adaptation and discriminative training techniques.

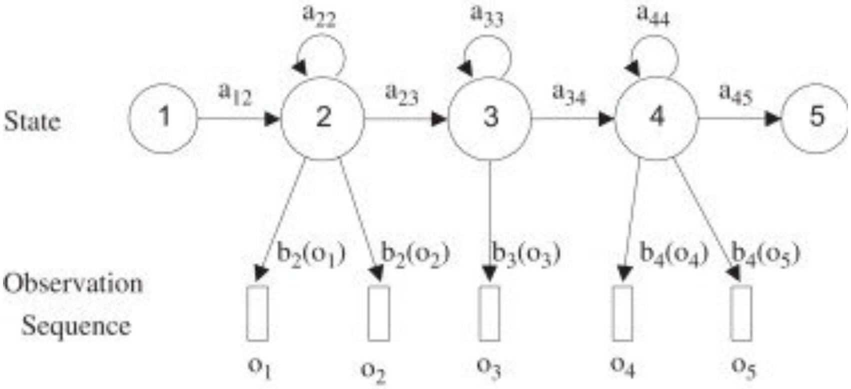


Figure 3 HMM/GMM acoustic model.

HMM/SGMMs

Although HMM/GMMs were a mainstream in AM, such the framework was not found appropriate for domain transfer (i.e., their adaptation to for instance low-resourced scenarios where the model is supposed to perform well also in case of relatively small amounts of training data). An interesting approach to overcome problems with requirements of large amounts of training data was presented by SGMMs - a new acoustic modeling scheme based on Sub-space Gaussian Mixture Model (SGMM), proposed in [Pov+10] (see Figure 4). SGMMs demonstrated their large potential to benefit from available data from well-resourced domains to improve recognition performance of the target domain.

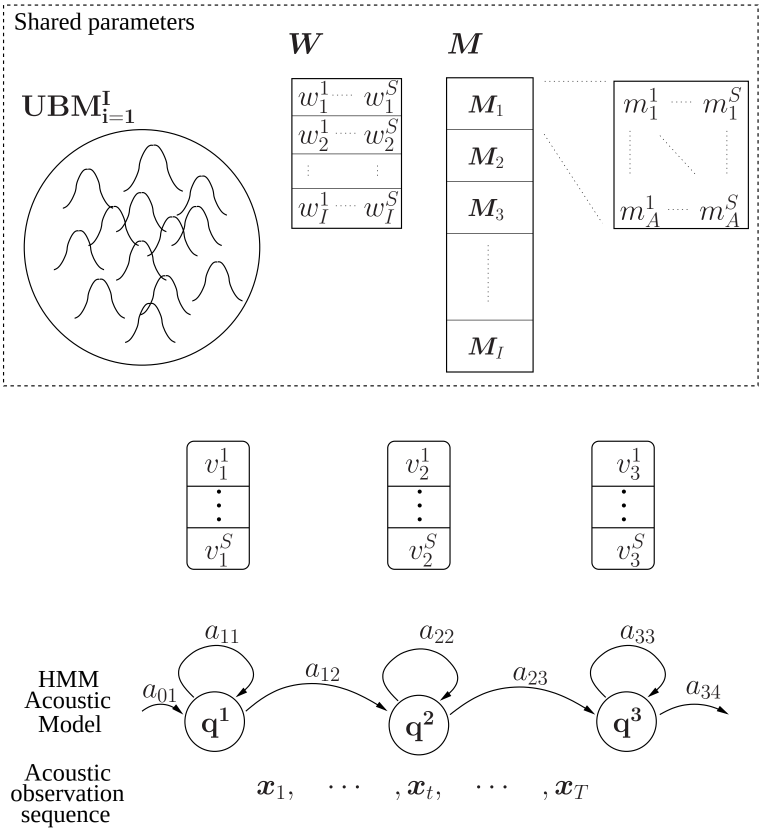


Figure 4 SGMM acoustic model (source: [Ims+13b]).

This characteristic was also shown for language transfer [Bur+10]. Compared to other (multilingual) techniques, such as traditional ones exploiting universal phone models to allow for training acoustic models from many languages [Lin+09], SGMM (as well as other models mentioned later such as KL-HMM and multi-lingual Tandem) can utilize a target phone set thus representing much simpler procedure. An example of application of SGMMs for language transfer learning is given in Section 2.4.1.

2.2.2 Hybrid models

Hybrid models for acoustic modeling in ASR were also proposed in 90's. They typically exploit an HMM framework, but the modeling of feature observations is done using artificial neural networks which are usually trained to estimate phone (or any other sound units) posteriors while also including a temporal context.

HMM/ANNs

Competitive acoustic modeling technologies to the generative HMM/GMMs consider Artificial Neural Networks (ANNs), allowing to discriminatively train the acoustic classifier. First ANN models employed Multi-Layer Perceptrons (MLPs) [BM94], also combined with Markov chains, referred to as hybrid HMM/ANNs. HMM/ANN were usually trained to estimate sound-unit (phone) posteriors based on the input speech features. Eventually, these posteriors estimated by ANNs are then transformed to scaled likelihoods and used directly as output probabilities in the HMM topology (i.e. replacing likelihoods estimated by GMMs in HMM/GMMs).

Tandem

Among interesting approaches exploiting disriminatively trained acoustic classifiers were "Tandem" models (see Figure 5). Such an acoustic model combined spectral features (MFCCs) with another set of features derived from phone-classification MLPs. Rather than interpreting the outputs as phone posteriors, they were subject to a logarithm transformation and dimensionality reduction, and (used in a combination with MFCCs) as final input features in HMM/GMM architecture [HES00].

HMM/DNNs

Most recent hybrid AM approaches applied for automatic speech recognition use modern Deep Neural Networks (DNNs) and allow to exploit large temporal contexts (e.g., in the form of time-delay neural architectures). Similar to HMM/ANN, these hybrid HMM/DNN systems use the DNNs to estimate posterior probabilities of usually context-dependent HMM states (see Figure 6).

KL-HMMs

As illustrated in Figure 7, KL-HMM is a particular form of HMM in which the emission probabilities are parametrized by a categorical distribution, i.e., a multinomial distribution from which only one sample is drawn. In contrast to Tandem that uses Gaussian mixtures and therefore needs the post-processed features, the categorical distributions can directly be trained from phone class posterior probabilities (e.g. by ANNs or DNNs).

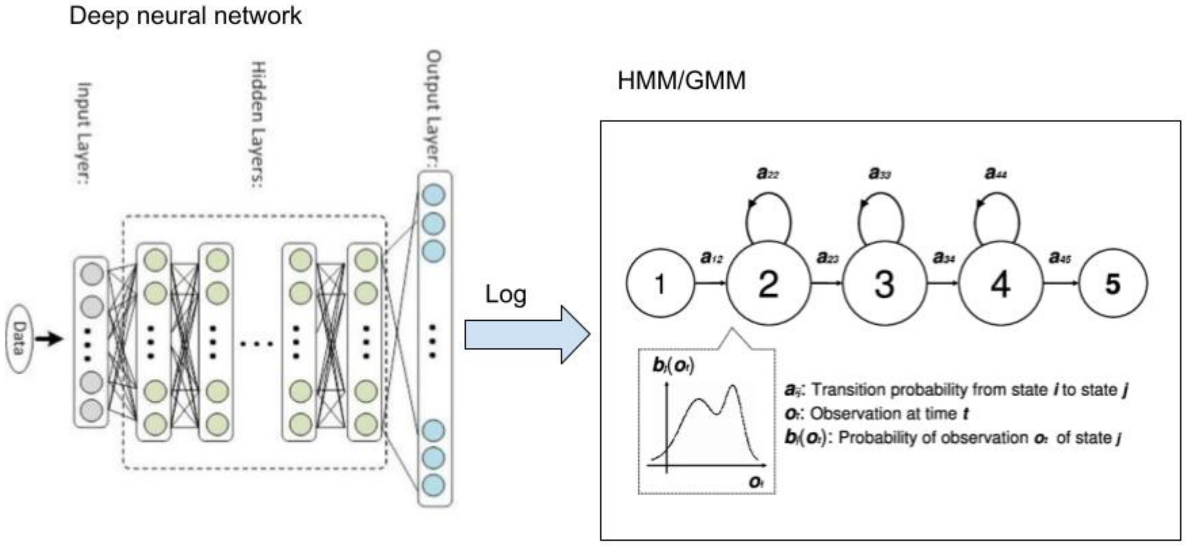


Figure 5 Tandem acoustic model.

More particularly, a term of discrimination information is nowadays referred to as the Kullback–Leibler (KL) distance (or divergence as it is not a metric), defined by [KL51]. In [ABM08], authors proposed multiple KL divergence based local scores for KL-HMM training and decoding. In other studies, the symmetric variant of the KL divergence was used. However, recently it was found that the asymmetric KL divergence gives better performance and increases the robustness of ASR systems. This is also intuitively reasonable in that the underlying acoustic modeling problem is not symmetric since we observe the posterior features and train the categorical distributions.

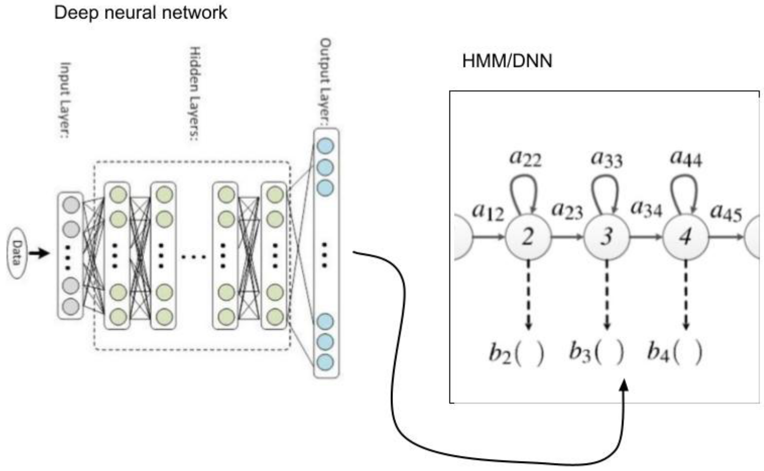


Figure 6 HMM/DNN acoustic model.

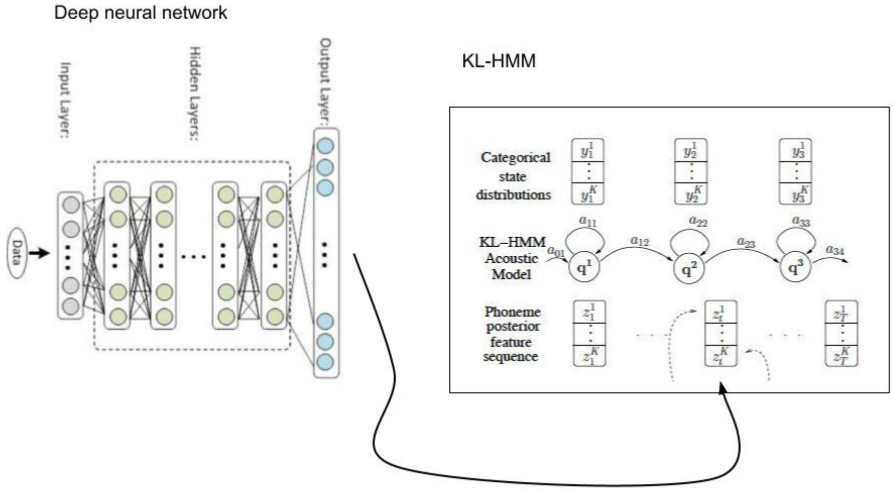


Figure 7 KL-HMM acoustic model.

2.3 Language modeling

Apart from acoustic modeling, language models are also among important building modules of ASR systems. In this section, we will first give a brief introduction to language modeling technology.. Second, as presented in Section 2.4.3, the language models also offer a transfer learning capability (e.g., for a new domain), similar to acoustic models. As part of Section 4 related to natural language understanding, we will present our work on language model adaptation (as supervised and unsupervised approach),as well as the work on using recurrent neural network based LMs in ASR.

In addition to automatic speech recognition, LMs are widely used in many other fields of natural language processing. The principal objective of LMs is to assign a probability to an utterance, e.g. a sentence, estimating how likely it is to observe this utterance in the language. As such, the LM performance has a significant impact on the performance of the ASR. In case of ASR, the most commonly used approach is the n-gram – a purely statistical approach to estimate probabilities for new utterances by collecting statistics from a training text corpus. If we consider the same language, using a larger text corpus or increasing the model order typically improves LM performance, but also increases its size. N-gram makes a Markov assumption, i.e., the probability of observing a specific word in a sentence only depends on the last $n-1$ observed words. In earlier times, mostly bi-gram models $(n=2)$ have been used, whereas nowadays language model orders of $n=3$ (tri-grams), $n=4$ or even $n=5$ are common [Man11]. Although the simplicity of an n-gram language model obviously cannot possibly convey the complexity of real natural language and research into more complex LM types has been conducted for decades [Jel91], n-gram LMs persist as a very popular type of language model used.

In case of n-grams, one of the most typical problems is a balance weight between infrequent grams (for example, if a proper name appeared in the training data) and frequent grams. Also, entities not seen in the training data will be given a probability of 0.0 without “smoothing” method. This method is therefore necessary to smooth the probability distributions by also assigning non-zero probabilities to unseen words or n-grams. Kneser–Ney smoothing [NEK94] is considered among the most effective methods, primarily used to calculate the probability distribution of n-grams in a document based on their histories.

Advanced LMs

(Paper also for consideration in this section: The Kaldi Speech Recognition Toolkit, Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer and Karel Vesely, in: IEEE 2011 Workshop on Automatic Speech Recognition and Understanding, Hilton Waikoloa Village, Big Island, Hawaii, US, IEEE Signal Processing Society, 2011 [Pov+11])

(Paper also for consideration in this section: STACKED NEURAL NETWORKS WITH PARAMETER SHARING FOR MULTILINGUAL LANGUAGE MODELING, Banriskhem Khonglah, Srikanth Madikeri, Navid Rekabsaz, Nikolaos Pappas, Petr Motlicek and Hervé Bourlard, Idiap-RR-12-2019 [Kho+19])

Conventional ASR development tools (such as Kaldi [Pov+11]) use FST-based frameworks which in principle allow to deploy any language model that can be represented as a Finite State Transducer (FST)[3]. Neural based LMs (most typically Recurrent Neural Networks(RNNs)) are often used to re-score top ranked hypotheses obtained using conventional back-off n-gram LMs. Alternatively, the lattices generated with n-gram LMs may be directly re-scored. Recently, a LM based on Time Delay Neural Network (TDNN) architecture, in which the convolution is applied with respect to only the past time steps to avoid any leakage from the future time steps, was proposed. TDNN based LMs were shown to have lower perplexity than RNN LMs [Kho+19]. RNNs are also further analysed in Section 4.3, to be used for a direct decoding (i.e., replacing a two-pass ASR where RNNs are used to re-score word recognition hypotheses).

2.4 Domain and language transfer learning for acoustic and language modeling

In this section, we will review algorithms used for both domain and language transfer applied in ASR. The focus will mainly be on transfer learning capabilities of acoustic models, nevertheless, a short section will also be devoted to typical approaches applied for LM transfer learning.

By transfer learning, we principally mean the model's ability to adapt to a new domain or language. There are several needs for this type of approaches:

- Adaptation to low resource languages: Many languages in the world can be considered as low-resourced, i.e., there are not sufficiently large corpora available for developing robust and accurate acoustic or language models for a given language. The same may apply for different accents (e.g. French English or German English), or dialects (e.g. Swiss German).
- Adaptation to challenging domains: Even in case of developing the ASR for viable languages, there are typical problems with robustness of the ASR systems for specific domains. Among many examples, we can consider analysis of multi-party interactions (e.g. remote meetings), or, as it will be particularly addressed here, recognition of air-traffic communication, where there are not many data resources available to develop sufficiently robust systems.

[3] https://en.wikipedia.org/wiki/Finite-state_transducer

This section will review our past works related to both aforementioned problems.

2.4.1 Cross-lingual acoustic modeling

(Relevant paper: Using out-of-language data to improve an under-resourced speech recognizer, David Imseng, Petr Motlicek, Hervé Bourlard and Philip N. Garner, in: Speech Communication, 2013 [Ims+13b])

(Paper also for consideration in this section: Exploiting foreign resources for DNN-based ASR, Petr Motlicek, David Imseng, Blaise Potard, Philip N. Garner and Ivan Himawan, in: EURASIP Journal on Audio, Speech, and Music Processing(2015:17), 2015 [Mot+15])

(Paper also for consideration in this section: Multilingual Deep Neural Network based Acoustic Modeling For Rapid Language Adaptation, Ngoc Thang Vu, David Imseng, Daniel Povey, Petr Motlicek, Tanja Schultz and Hervé Bourlard, in: Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing, Florence, pages 7639-7643, IEEE, 2014 [Vu+14])

One of the important tasks in ASR is to address cross-language transfer, specifically for languages with low data resources. In a practical case this means the models developed for well-resourced language (e.g., Dutch) can be adapted (or fine-tuned) in a cross-lingual manner to the target language (e.g., Afrikaans) as considered in our paper [Ims+13b]). More particularly, this paper from 2012 considered a scenario where out-of-language data (i.e., audio data available from another language, different to the target language) can boost the ASR performance of the within language having only limited amounts of data for training/development. The study investigated both the generative (HMM/GMM) and discriminative (HMM/ANN) acoustic models and their extended versions for cross-lingual transfer.

Cross-lingual Tandem and KL-HMMs

The MLP for acoustic modeling is typically trained to estimate phone class posterior probabilities given the speech feature vectors. For the domain or language transfer, the model is usually built on a large set of out-of-domain data (or language) and thus called an “auxiliary” MLP. The approaches which exploit MLP to generate posterior probabilities are denoted as feature-level based approaches, in this section further used for model adaptation.

More specifically, once the MLP is trained: (i) we consider a sequence of T acoustic feature vectors $X = \mathbf{x}_1, ..., \mathbf{x}_T$, namely Perceptual Linear prediction (PLP) speech features [Her89], extracted from within-language data, and (ii) the phone class posterior sequence $Z = \mathbf{z}_1, ..., \mathbf{z}_T$ is then estimated with the previously trained auxiliary MLP. To estimate $\mathbf{z}_t = (z_t^1, ..., z_t^K)^T$, we consider also a temporal context of $\mathbf{x}_t$ features. Further, two modeling approaches were used to model the class posterior sequence [Ims+13b]:

- Tandem: as already shown in Figure 5, it uses GMMs for estimating emission probabilities of HMMs [HES00] (i.e., HMM states q^d (where $q^d : d \in 1, ..., D$) are associated with the target language (D is equal to total number of HMM states in the model)). To model the emission probabilities with Gaussians, the posterior features $\mathbf{z}_t$ are usually post-processed (decorrelated with a principal component analysis (PCA)). The transformation matrix can be estimated on within-language data. Usually, the resulting feature vector $\mathbf{r}_t = (r_t^1, ..., r_t^L)^T$ has a reduced dimensionality L. The approach is in details visualised in Figure 8.

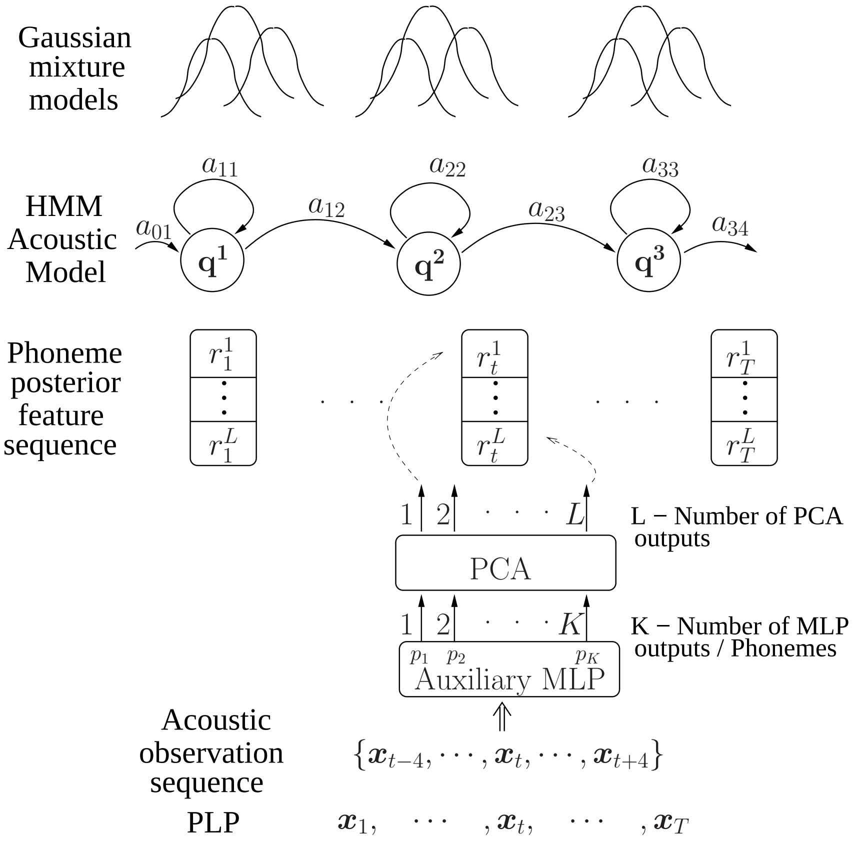

Gaussian mixture models
HMM Acoustic Model
a_{11}
a_{22}
a_{33}
a_{01}
a_{12}
a_{23}
a_{34}
$\mathbf{q^1}$
$\mathbf{q^2}$
$\mathbf{q^3}$
Phoneme posterior feature sequence
r_1^1
r_1^L
r_t^1
r_t^L
r_T^1
r_T^L
1 2 · · · L
L − Number of PCA outputs
PCA
1 2 · · · K
K − Number of MLP outputs / Phonemes
p_1 p_2 p_K
Auxiliary MLP
Acoustic observation sequence
$\{\boldsymbol{x}_{t-4}, \cdots, \boldsymbol{x}_t, \cdots, \boldsymbol{x}_{t+4}\}$
PLP
$\boldsymbol{x}_1, \cdots, \boldsymbol{x}_t, \cdots, \boldsymbol{x}_T$

Figure 8 Tandem - the emission probabilities of the HMM states are modeled with Gaussian mixtures and the MLP output is postprocessed (source: [Ims+13b]).

- KL-HMM: as described in Section 2.2.2, KL-HMM is a particular form of HMM in which the emission probability of state q^d is parametrized by a categorical distribution $\mathbf{y}_d = (y_d^1, ..., y_d^K)^T$, where K is the dimensionality of the features. Unlike Tandem that uses GMM (with the post-processed features $\mathbf{r}_t$), the categorical distributions can directly be trained from $\mathbf{z}_t$. For acoustic modeling of ASR, [AVB07] proposed multiple KL divergence based local scores for KL-HMM training and decoding. KL-HMM is visualised in Figure 9.

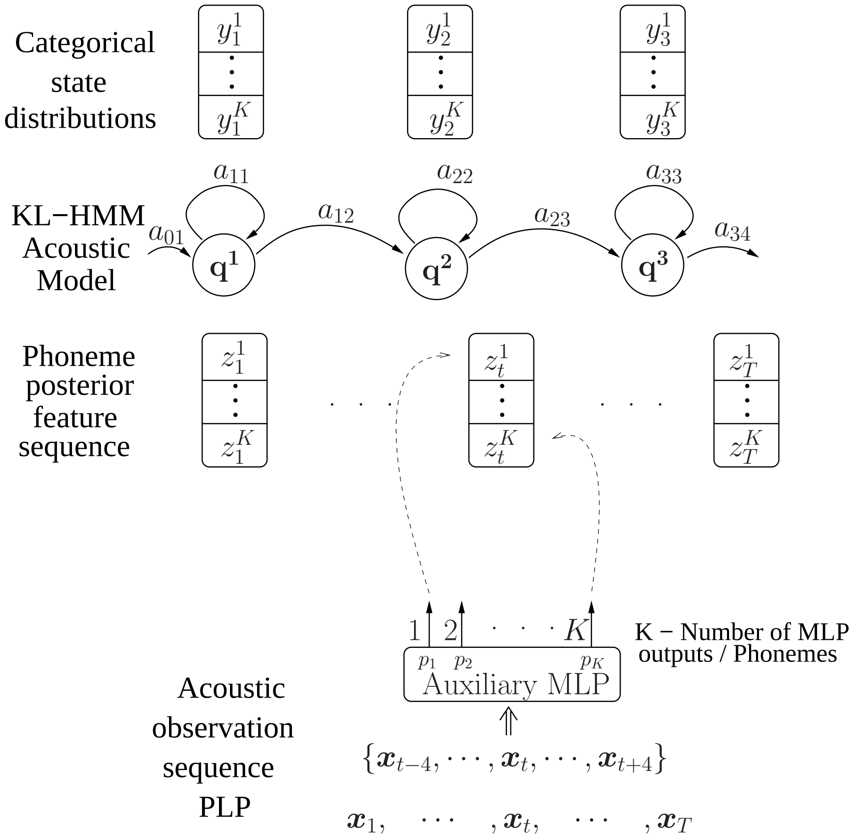

Categorical state distributions
y_1^1
y_1^K
y_2^1
y_2^K
y_3^1
y_3^K
KL−HMM Acoustic Model
a_{01}
a_{11}
a_{12}
a_{22}
a_{23}
a_{33}
a_{34}
$\mathbf{q^1}$
$\mathbf{q^2}$
$\mathbf{q^3}$
Phoneme posterior feature sequence
z_1^1
z_1^K
z_t^1
z_t^K
z_T^1
z_T^K
1 2 · · · K
K − Number of MLP outputs / Phonemes
p_1 p_2 p_K
Auxiliary MLP
Acoustic observation sequence PLP
$\{\boldsymbol{x}_{t-4}, \cdots, \boldsymbol{x}_t, \cdots, \boldsymbol{x}_{t+4}\}$
$\boldsymbol{x}_1, \cdots, \boldsymbol{x}_t, \cdots, \boldsymbol{x}_T$

Figure 9 KL-HMM - the emission probabilities are modeled with categorical distributions and the MLP output can directly be used (source: [Ims+13b]).

Cross-lingual SGMMs

A complementary approach to the feature-level adaptation is the model-level adaptation allowing to exploit out-of-language data directly on the acoustic model level to perform cross-lingual transfer and eventually improve ASR performance on target language. In our past work, we specifically experimented with Subspace GMMs (i.e., HMM/SGMMs), already introduced in Section 2.2.1. Similar to feature level, HMM state distributions associated with the target language are estimated. The transition probabilities are fixed and the emission probabilities are modeled using probability density function in an SGMM manner.

Mathematically, the SGMM model is described in [Pov+10], where the emission probabilities of each context-dependent HMM-state q^d are modeled by GMM. Each HMM-state is parametrized by a vector $\mathbf{v}_d$. The parameters $\mathbf{M}$ and $\mathbf{W}$ are globally shared and are used to derive the means and mixture weights representing the given HMM state. Graphical interpretation of SGMMs as an acoustic model was given in Figure 4.

The results presented in [Ims+13b] have shown that out-of-language data (in this experiment Dutch speech) can significantly improve speech recognition on Afrikaans (i.e., target language represented by only a small data set).

Cross-lingual DNNs

More recent approach published in 2015 [Mot+15], which is already using modern Deep Neural Networks (DNN) as an acoustic model employed in HMM framework, performs a cross-lingual transfer through a condition-specific layer. The idea is similar to multi-lingual DNN approaches (see Section 2.4.2) in which hidden layers of DNNs are shared across multiple (auxiliary) languages while the output layers are made language-specific. The adaptation procedure is graphically visualised in Figure 10. Starting with a DNN model trained using out-of-language data (in this case French ESTER database), the output layer is replaced by a new layer in which we randomly initialise the $\mathbf{W}_L$ which is the matrix of connection weights between the layer $L-1$ and the output layer L. The network is then retrained using in-language data (in this case French MP-FR database) which most closely matches the evaluation set.

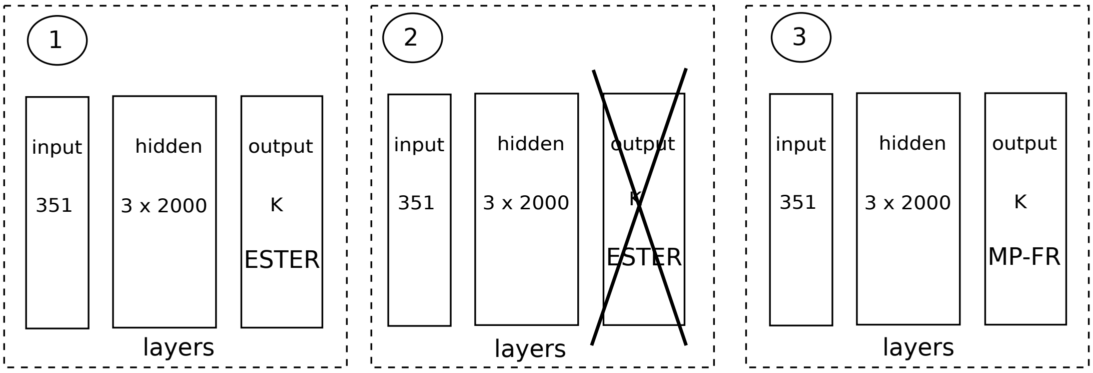


Figure 10 Cross-lingual adaptation in DNN (source: [Mot+15]).

Previous cross-lingual adaptation algorithms have considered, although different languages or dialects, the same phone sets shared across them. In case the phone sets are largely dissimilar, other alternative approaches in cross-lingual (or multi-lingual) transfer learning need to be considered, such as: (i) use of merged universal phoneset based on the International Phonetic Alphabet (IPA) chart[4], i.e. the same IPA symbols are merged across languages, or (ii) a universal phoneset without merging strategy. Our work presented in [Vu+14] (on ten different languages from the Globalphone database) investigated the effect of IPA based phoneme merging on the multi-lingual DNN and its application to new languages.

2.4.2 Multi-lingual acoustic modeling

(Relevant paper: Impact of deep MLP architecture on different acoustic modeling techniques for under-resourced speech recognition, David Imseng, Petr Motlicek, Philip N. Garner and Hervé Bourlard, in: Proceedings of the IEEE workshop on Automatic Speech Recognition and Understanding, 2013 [Ims+13a])

[4] https://en.wikipedia.org/wiki/International_Phonetic_Alphabet

Multi-lingual acoustic modelling shares the acoustic data across multiple languages to cover as much as possible the contextual variations of the considered languages. Unlike cross-lingual acoustic modeling (in details presented in Section 2.4.1) which envisaged a language transfer from one (source) to another (target) language, multi-lingual approaches consider multiple source languages during the development.

In our preliminary experiments [Ims+13a], we used a simplistic approach where data from several languages are joined (shared) for training the acoustic model. One way to achieve such data sharing is to define a common phonetic alphabet across all languages. As already presented above, this common phone set can be either derived in a data-driven way, or obtained from the IPA.

Multi-lingual Tandem, KL-HMMs and SGMMs

An extension of previous cross-lingual AM approaches (Tandem, KL-HMMs and SGMMs) assumed a common phone set (i.e., the set is similar for both languages) thus allowing to train an MLP jointly using both languages (Dutch and Afrikaans in that case). Unlike a cross-lingual approach described in Section 2.4.1 where the MLP was trained only using out-of-language data, this system uses both out-of-domain and in-domain data for MLP training.

Multi-task adaptation

Multi-task learning (adaptation) has led to successes in many applications of machine learning [Rud17], including automatic speech recognition [DHK13]. In case of multi-lingual AM training, it is among the most powerful approaches especially for low-resourced languages.

ASR for low-resource languages is often developed by adapting a pre-trained model to a target language. Similarly to previous approaches, this type of adaptation uses auxiliary speech data from other languages in addition to the target-language data. However unlike previous case where the specific layers in the DNN were adapted using in-language data (also called “single-task adaptation”), “multi-task adaptation” employs a similar strategy by adapting to the multiple languages at the same time (despite our interest being in target language). Multi-task adaptation has several rigorously proven advantages. Two important advantages that are often considered in AM training are (i) implicit data augmentation and (ii) ability to reduce the risk of over-fitting. Figure 11 illustrates the difference between single-task and multi-task training. Well documented set of experiments comparing multi-lingual approaches (such as using IPA or multi-task adaptation) while using deep neural nets was performed in [TGB17].

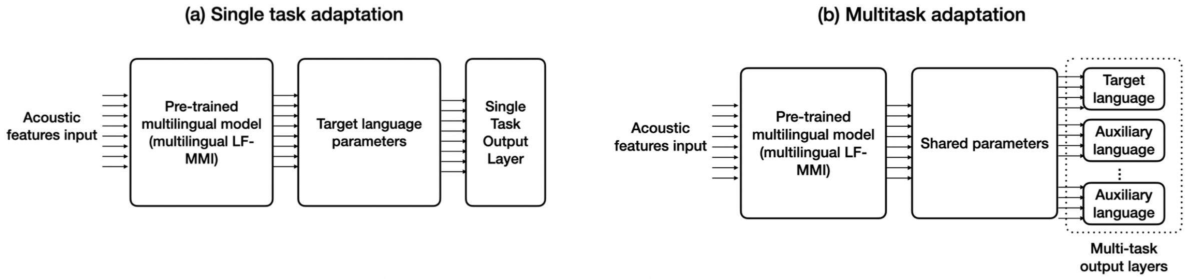


Figure 11 Single-task and multi-task adaptation for acoustic modeling (source: [MMB21]).

2.4.3 Domain adaptation and multi-lingual training of LMs

LM domain adaptation

Although only briefly discussed here, language model adaptation plays an important role (and complements with a significant boost of performance an acoustic model adaptation) when transferring ASR systems to be deployed for new (unseen) domains. As already mentioned for acoustic modeling, there are applications where the specific domain of data needs to be improved for successful application of ASR. One such application was in the MATERIAL programme[5]. In MATERIAL, targeted by both BUT[6] and Idiap[7], the ASR for low-resource languages was researched for document retrieval and summarization purposes.

Linear combination – One of relatively simple but still preferable approaches, successfully deployed in production ASRs (not only in domains with limited amount of in-domain training data such as MATERIAL), relies on model combinations [Hsu07]. In the case of LM, these approaches include improving the estimation of the underlying probability distributions. While many of these approaches involve the combination of multiple n-gram LMs, most existing works only evaluate their performance using simple linear interpolation [Jel80].

From a practical point of view, linear interpolation first trains individual n-gram LMs separately for each training corpus (e.g., from out-of-domain and in-domain text). Given the resulting set of n-gram LMs, it computes the weighted average of the component model probabilities, while the interpolation weight is typically tuned to optimize the development set perplexity. To increase its efficiency, an approximation is often applied on the final interpolated model that constructs a single n-gram back-off model where the probability for all observed n-grams is represented by the weighted average of the component model probabilities [Sto02].

[5] https://www.iarpa.gov/index.php/research-programs/material
[6] https://www.fit.vut.cz/research/project/1140/.en
[7] https://www.idiap.ch/en/scientific-research/projects/SARAL

Neural networks – Successful approaches for domain transfer often consider Neural Networks (NNs). NN-based LMs are widely used for re-scoring the N-best list of word recognition hypotheses obtained from the decoding based on n-grams [Mik+10]. This is usually called the second pass decoding. There were also attempts to use the neural networks for first pass decoding (which will be discussed later in Section 4.3).

Multi-lingual LM

For many domains such as conversational speech, there is less availability of data, thus in-language domain adaptation usually does not provide sufficient performance (e.g., also pointed out by many works from MATERIAL programme). For such domains, multi-lingual NNLMs sharing parameters across multiple languages may be of large interest. These models aim to address these data sparsity issues [Rag+16].

One of our past work proposed multi-lingual architecture consisting of a stacked NN model, where the first layer is language-specific and the second one is shared across multiple languages. In addition, and in contrast to [Rag+16], every language has a separate input and output layer and hence a separate loss function. The overall loss in our proposed approach was the weighted sum of per-language loss values, used to optimize the whole network through back-propagation [Kho+19].

2.5 Sequence discriminative training for AM

Significant improvements in acoustic modeling have also been obtained by exploiting sequence-discriminative training approaches, first applied on generative HMM/GMM modeling [Bah+86] [Pov+08].

Typical objective function for estimating the model parameters in HMM based speech recognition systems is Maximum Likelihood Estimation (MLE). If we assumed that the speech matched the statistics expected by an HMM and we had access to an infinite training set, the global maximum likelihood estimate would be optimal in the sense that it is unbiased with minimum variance [WP00]. However, this is usually not the case. It has been shown that alternative discriminative training schemes such as the most popular Maximum Mutual Information MMI estimation provide generally better ASR performance.

For applying MMI training for acoustic model in ASR, typical approach is that the derivatives of MMI objective function are computed from two sets of posterior quantities: (i) for the numerator graph, specific to each utterance related to the alignment (i.e. with text transcript) and (ii) for the denominator graph, which represents all possible word sequences and which is the same for all utterances. In Kaldi ASR framework [Pov+11], the Finite State Acceptor (FSA) format is used to store both of them (with labels on arcs, not states).

The theory for sequence-discriminative training of neural networks was also developed quite early in 90's [BD91], where the posterior probabilities are same as the numerator and denominator occupancies used in discriminative training of HMM/GMM systems. Later, it was also pointed out that sequence-discriminative training of NNs can take advantage of the lattice-based computations that were routinely used for HMM/GMM systems. Very recently, the Lattice-Free Maximum Mutual Information (LF-MMI) framework has shown to have superior performance compared to the conventional Cross-Entropy (CE) training of DNNs [Had+18]. Similar to HMM/GMMs, the MMI cost function uses the numerator graph modelling the observed speech feature sequence based on ground-truth transcript and the denominator graph computing the probability over all possible sequences. The latter enforces the discriminative property in the training shown to be useful for AM development.

2.5.1 Extension to multi-lingual acoustic modeling

(Relevant paper: Lattice-Free Maximum Mutual Information Training of Multilingual Speech Recognition System, Srikanth Madikeri, Banriskhem Khonglah, Sibo Tong, Petr Motlicek, Hervé Bourlard and Daniel Povey, in: Proceedings of Interspeech, pages 4746–4750, ISCA, 2020 [Mad+20a])

(Relevant paper: Multitask adaptation with Lattice-Free MMI for multi-genre speech recognition of low resource languages, Srikanth Madikeri, Petr Motlicek and Hervé Bourlard, in: Proceedings of Interspeech, 2021 [MMB21])

Combination of multi-lingual modeling and sequence-discriminative training is nowadays considered as state-of-the-art framework to train large-scale hybrid acoustic models. As already mentioned in Section 2.4.2, in case of single-task approach, the multi-lingual resources are combined by merging the phoneme sets from all languages either using a universal phone set such as IPA, or by combining acoustic units (phones) across languages. If sequence-discriminative training is applied in any of these cases, a universal denominator graph is shared across all languages during training, as shown also for LF-MMI for instance by [TGB19]. However, when combining acoustic units for multi-lingual training in single-task approach, the output NN layer size increases rapidly with number of languages, which may become impractical during decoding. Alternately, multi-task training (already presented in Section 2.4.2) solves this issue by separating the output layers of languages so that during decoding only the output relevant to the language is used. An added advantage during training is that the cost function can be computed faster as its complexity depends on the number of states in the denominator.

In single-task case when implemented for LF-MMI multi-lingual AM, the configuration provides a choice of using language-specific (i.e. trained with data from all languages), or language-independent denominator (i.e., trained with data from only one language, which is equivalent to training mono-lingual AMs). When using language-specific denominators, the cost function changes: each denominator graph is built from the language-specific phone language model (the same as that used in mono-lingual LF-MMI training). Gradients for language-dependent layers are computed and updated for each minibatch. Using back-propagation, the shared parameters are then updated. The overall cost-function is the weighted sum of all language-dependent cost-functions.

In multi-task case, the language-independent denominator is applicable only. The work was proposed in [Mad+20a] and the code implementation was made available as a part of Kaldi [Pov+11][8].

2.5.2 Extension to multi-lingual acoustic modeling and language transfer

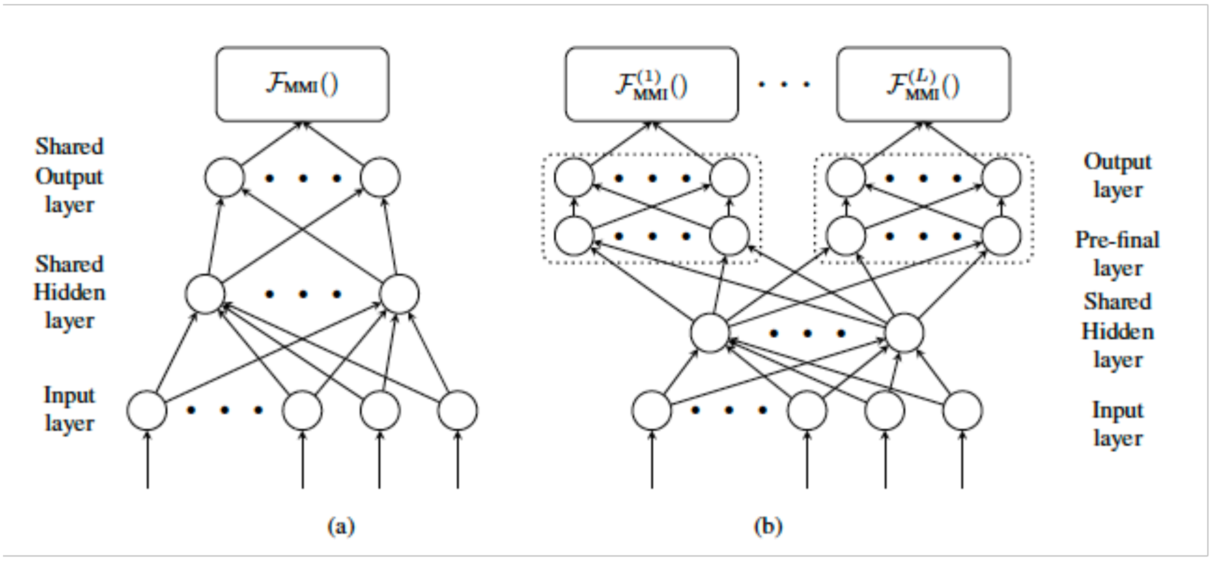


Figure 12 LF-MMI training for multi-lingual acoustic modeling (source: [Mad+20a]).

In addition to multi-lingual training, we also recently presented a work on (multi-task) language transfer [MMB21] while employing sequence discriminative training. The reason for combining both multi-task and sequence discriminative training approaches is that when adapting pre-trained acoustic models to low-resource languages, it can be observed that despite heavy regularization (e.g., high dropout rates), the model performance usually saturates. To avoid such saturation, the regularization is often used, which in this case is done by presenting other languages for training (see Figure 11). The work was performed as part of MATERIAL pro-

[8] egs/babel multilang/s5d/local/chain2/run tdnn.sh

gramme thus many languages were considered assuming relatively small development data is available. The multi-lingual model in this work [MMB21] was developed on 18 languages with LF-MMI criterion. For multi-task training, Kazakh, Pashto and Farsi languages were used. The first two languages were also part of the 18 languages used for multi-lingual training while Farsi was an unseen language. The work also applied single-task training for comparison.

Eventually, the multi-task adaptation code was released[9] as part of the Babel multi-lingual recipe in Pkwrap [Mad+20b] to adapt both Kaldi and Pytorch [Pas+19] acoustic models trained with LF-MMI.

2.6 Semi-supervised training

In order to reach sufficient accuracy, the state-of-the-art ASR systems require large datasets for the development. The typical supervised training requires speech recordings with manual transcripts together with a collection of linguistic data resources for lexicon and language modeling. However, the data preparation can be very slow and costly. To avoid this, semi-supervised training can be of interest, as it can significantly reduce the data preparation time and cost by transcribing only a subset of the data while the rest of data is transcribed automatically. One of the works, also implemented as part of Kaldi, was developed at BUT [VHB13].

In our work, we mostly used methodology in which the transcribed data are used to build a seed model. The seed model is then used to decode untranscribed data and the resulting hypotheses represent ground-truth transcripts in further training. Typically, the data are selected according to some form of a confidence measure.

2.6.1 Data selection

(Relevant paper: Semi-supervised Learning with Semantic Knowledge Extraction for Improved Speech Recognition in Air Traffic Control, Ajay Srinivasamurthy, Petr Motlicek, Ivan Himawan, Gyorgy Szaszak, Youssef Oualil and Hartmut Helmke, in: Proceedings of Interspeech, 2017, Stockholm, Sweden, pages 2406-2410 [Sri+17])

(Paper also for consideration in this section: Contextual Semi-Supervised Learning: An Approach To Leverage Air-Surveillance and Untranscribed ATC Data in ASR Systems, Juan Zuluaga-Gomez, Iuliia Nigmatulina, Amrutha Prasad, Petr Motlicek, Karel Vesely, Martin Kocour and Igor Szoke, in: Proceedings of Interspeech, 2021 [Zul+21])

The automatically generated transcripts (along with transcribed speech) can be used as training data. However, these automatic transcripts will most probably be erroneous and those

[9] https://github.com/idiap/pkwrap/tree/master/egs/multilang/babel/

with most significant errors should be excluded from training, which is a problem often termed as “data selection”. Typically, data selection is done by assigning confidence scores to ASR outputs, so that high confidence transcripts (and corresponding utterances) can be selected for further training. In our past work, we explored two different data selection strategies: (i) word level confidences and (ii) concept and command level confidences (as part of the work on analysing air-traffic communication). Both data selection methods aim to utilize automatically transcribed data to provide additional training resources [Sri+17].

2.6.2 Semi-supervised training using LF-MMI

A simple approach to semi-supervised training in the LF-MMI framework (see Section 2.5 for more details about the framework) is to generate 1-best output as transcription for the unlabelled data. Also posteriors in the 1-best path can be used in the lattices generated during decoding as frame weights. The 1-best path is used as a numerator graph during semi-supervised training, where the supervised and unsupervised data are combined together. The confidence scores obtained from the LF-MMI system are often sparse thus to get informative measures for data selection or weighting, some post-processing needs to be applied.

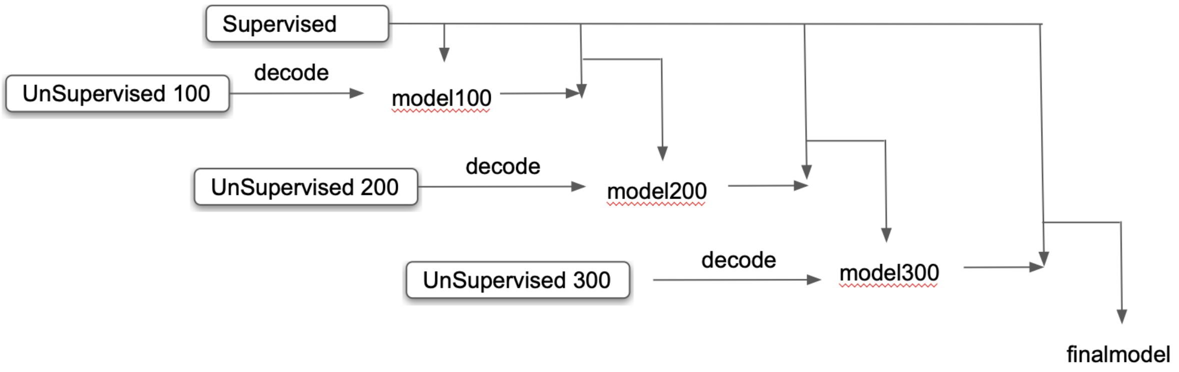


Figure 13 Incremental approach for semi-supervised training. Model100 means acoustic model developed using 100 hours of untranscribed data, etc.

2.6.3 Incremental semi-supervised training

(Relevant paper: INCREMENTAL SEMI-SUPERVISED LEARNING FOR MULTI-GENRE SPEECH RECOGNITION, Banriskhem Khonglah, Srikanth Madikeri, Subhadeep Dey, Hervé Bourlard, Petr Motlicek and Jayadev Billa, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020 [Kho+20])

In order to improve the quality of transcriptions produced for the untranscribed data, recently we proposed a simple method to generate and update labels without any change to the core semi-supervised training framework being employed. The work was motivated based on the observation that semi-supervised training can improve the acoustic model even with limited amounts of untranscribed data. As seen in Figure 13, our approach divides the entire untranscribed dataset into several equalsized parts (i.e., per 100 hours) and starts the semi-supervised training with only one part. While there exist many ways to divide the data, in this work we have considered closely matching the amount of supervised data to our split-size.

Enumerating each split from $1...n$, we run n training iterations. In the i-th iteration, splits $1...i$ are used as the untranscribed set for training. As shown in Figure 13, in each iteration we use the previous model as the seed for a new iteration of semi-supervised training from scratch. The data used for each iteration includes the supervised set, all the portions of the unsupervised set used in the last iteration and one unused sub-set for the current iteration. In doing so, we are continuously improving the seed model on the domain of the untranscribed data. We note that this data scheduling strategy, however, is computationally intensive since it involves multiple decodes of the data.

2.6.4 Semi-supervised training for language modeling

Previous sections were related to training of acoustic models while using the data without manual transcripts. In order to develop a good quality production system, the LM also requires large data for training.

Our typical approach in this case (i.e., specifically in the case where small amounts of transcripts are available), builds a second LM using the textual resources crawled from the internet. More details will be given in Section 4.2. Finally, linear interpolation is typically applied combining LM built from available manual transcripts with the one built from crawled textual resources.

2.7 Self-supervised acoustic model training

(Paper also for consideration in this section: End-to-End Accented Speech Recognition, Thibault Viglino, Petr Motlicek and Milos Cernak, in: Proceedings of Interspeech, ISCA, Graz, Austria, pages 2140-2144, 2019 [VMC19])

Similar to semi-supervised training, self-supervised training methods aim to learn powerful acoustic representations from untranscribed audio data. It has been shown that such acoustic models can later be adapted using supervised data to achieve state-of-the-art performance for ASR) while greatly reducing the amount of transcribed training data which is both expensive and time-consuming to obtain.

In our works related to rapid development of ASR for low-resource languages, we specifically considered wav2vec 2.0 [Bae+20], which learns representations from raw audio data using contrastive learning. Typical model, which was found of large interest, was trained on English read speech (i.e., 1000 hours of unsupervised Librispeech data [Pan+15]) and later adapted on a 100 hour supervised subset of Librispeech data to achieve state-of-the-art performance. The past works only considered Connectionist Temporal Classification (CTC) [Gra+06] for acoustic model training. Our recent work done as part of MATERIAL programme [VMB21]:

- (i) compared the effect of sequence discriminative training criterion for supervised adaptation and showed that fine-tuning the wav2vec 2.0 model with end-to-end version of LFMMI and CTC criterion yields roughly similar performances;
- (ii) the wav2vec 2.0 model (concretely XLSR-10 model [Con+20]) was further adapted on out-of-domain conversational speech and on cross-lingual data and achieved ASR results showed that the wav2vec 2.0 pretraining provides significant gains over the models trained only with supervised data.

One of our recent approaches used multi-task training and accent embedding in the context of end-to-end ASR trained with the connectionist temporal classification loss [VMC19].

Our very new work from spring 2022 targeted the scenario for the case when the data substantially differs between the pre-training and downstream fine-tuning phases (i.e., domain shift). We analyzed the robustness of wav2vec2.0 and XLSR models on downstream ASR for a completely unseen domain, i.e., air-traffic control communications.

2.8 Application of ASR in air-traffic management

(Paper also for consideration in this section: Automatic Call Sign Detection: Matching Air Surveillance Data with Air Traffic Spoken Communications, Juan Zuluaga-Gomez, Karel Vesely, Alexander Blatt, Petr Motlicek, Dietrich Klakow, Allan Tart, Igor Szoke, Amrutha Prasad, Seyyed Saeed Sarfjoo, Pavel Kolcarek, Martin Kocour, Honza Cernocky, Claudia Cevenini, Khalid Choukri, Mickael Rigault and Fabian Landis, in: Proceedings of 8th OpenSky Symposium, OpenSky Network, pages 1-10, MDPI, 2020 [Zul+20a])

(Paper also for consideration in this section: Automatic Speech Recognition Benchmark for Air-Traffic Communications, Juan Zuluaga-Gomez, Petr Motlicek, Qingran Zhan, Rudolf Braun and Karel Vesely, in: Proceedings of Interspeech, pages 2297-2301, 2020 [Zul+20b])

Air-Traffic Control (ATC) is a very demanding task where one or several Air-Traffic Controllers (ATCos) plan, send, and execute commands via voice communications, in order to ensure the safety of the airplanes in a given space area. ATC communication is one of typical cases where ASR systems would significantly help controllers to improve their efficiency and possibly decrease their workload, allowing also to spot errors in spoken communication, etc. A graphical example of ASR in ATC is given in Figure 14.

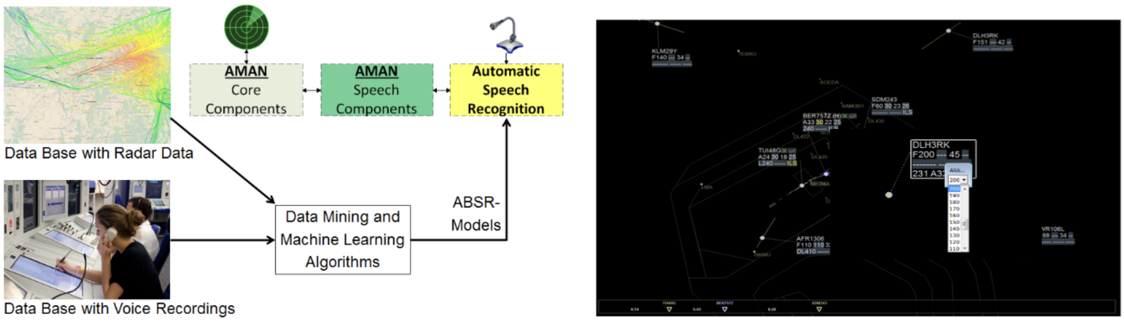
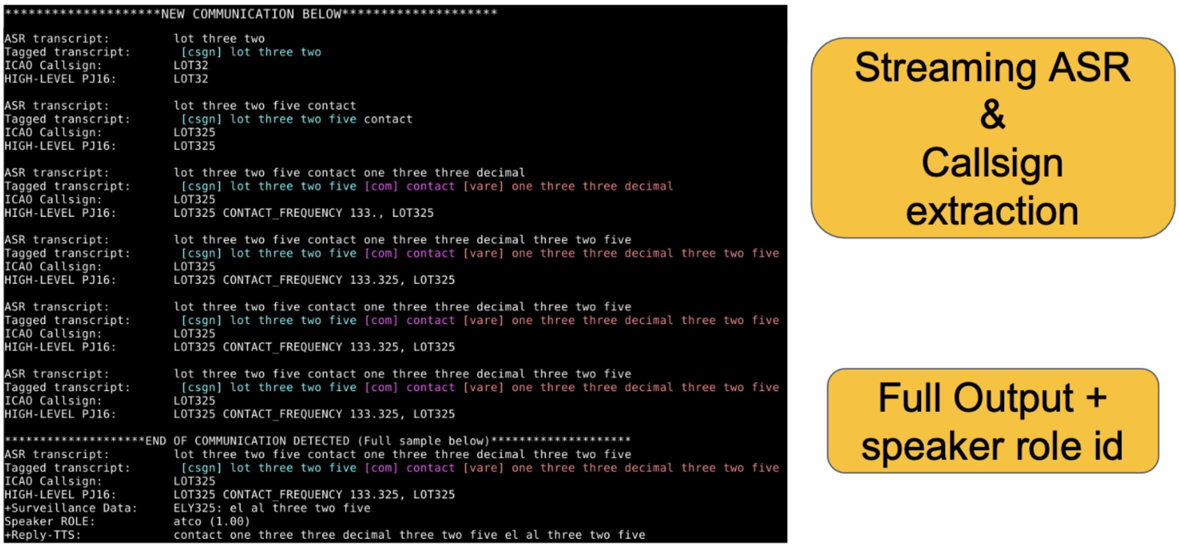


Figure 14 Overview of air-traffic management system incorporating automatic speech recognition.

Both Idiap and BUT have started collaborating on this topic already in 2016, through several projects. ATC communication currently relies on two approaches: (i) voice communication and (ii) voiceless communication through data links (also called CPDLC systems). One example of a CPDLC system is the Eurocontrol Link200+ [Eur12], which was expected to be deployed in all European airports by 2016. The idea is to transfer certain commands and orders through a human-machine interface, thus reducing the amount of spoken communication, but increasing the ATCos' workload. The International Civil Aviation Organization (ICAO) stated that "To minimize pilot head down time and potential distractions during critical phases of flight, the controller should use voice to communicate with aircraft operating below 10,000 ft above ground level"; hence, voice communications remains as the main way to exchange information and commands near airports. Recent research projects [Hol+15] and the ICAO have stated that air-traffic is expected to grow between 3% and 6% percent yearly at least until 2025 (i.e., estimated before COVID pandemic). The European Union (EU) with the aim of decreasing the ATCos' workload has invested resources into projects such as MALORCA (MAchine Learning Of speech Recognition models for Controller Assistance)[10], AcListant[11], and more recently ATCO2 (Automatic collection and processing of voice data from air-traffic communications)[12] and HAAWAII (HIGHLY AUTOMATED AIR TRAFFIC CONTROLLER WORKSTATIONS WITH ARTIFICIAL INTELLIGENCE INTEGRATION)[13], which have demonstrated various achievements by integrating spoken language understanding systems (including ASR: see a simplified output of the ATM based in Figure 15) on reducing the ATCos' workload [Hel+16], increasing the efficiency [Hel+17], and even offering better solutions in integrating contextual information, also in real time [Oua+15].

[10] http://www.malorca-project.de
[11] http://www.AcListant.de
[12] https://www.atco2.org
[13] https://www.haawaii.de



```
*******************NEW COMMUNICATION BELOW********************

ASR transcript:          lot three two
Tagged transcript:        [csgn] lot three two
ICAO Callsign:           LOT32
HIGH-LEVEL PJ16:         LOT32

ASR transcript:          lot three two five contact
Tagged transcript:        [csgn] lot three two five contact
ICAO Callsign:           LOT325
HIGH-LEVEL PJ16:         LOT325

ASR transcript:          lot three two five contact one three three decimal
Tagged transcript:        [csgn] lot three two five [com] contact [vare] one three three decimal
ICAO Callsign:           LOT325
HIGH-LEVEL PJ16:         LOT325 CONTACT_FREQUENCY 133., LOT325

ASR transcript:          lot three two five contact one three three decimal three two five
Tagged transcript:        [csgn] lot three two five [com] contact [vare] one three three decimal three two five
ICAO Callsign:           LOT325
HIGH-LEVEL PJ16:         LOT325 CONTACT_FREQUENCY 133.325, LOT325

ASR transcript:          lot three two five contact one three three decimal three two five
Tagged transcript:        [csgn] lot three two five [com] contact [vare] one three three decimal three two five
ICAO Callsign:           LOT325
HIGH-LEVEL PJ16:         LOT325 CONTACT_FREQUENCY 133.325, LOT325

ASR transcript:          lot three two five contact one three three decimal three two five
Tagged transcript:        [csgn] lot three two five [com] contact [vare] one three three decimal three two five
ICAO Callsign:           LOT325
HIGH-LEVEL PJ16:         LOT325 CONTACT_FREQUENCY 133.325, LOT325

*******************END OF COMMUNICATION DETECTED (Full sample below)*******************
ASR transcript:          lot three two five contact one three three decimal three two five
Tagged transcript:        [csgn] lot three two five [com] contact [vare] one three three decimal three two five
ICAO Callsign:           LOT325
HIGH-LEVEL PJ16:         LOT325 CONTACT_FREQUENCY 133.325, LOT325
+Surveillance Data:      ELY325: el al three two five
Speaker ROLE:            atco (1.00)
+Reply-TTS:              contact one three three decimal three two five el al three two five
```

Figure 15 Output of ASR system developed for air-traffic management.

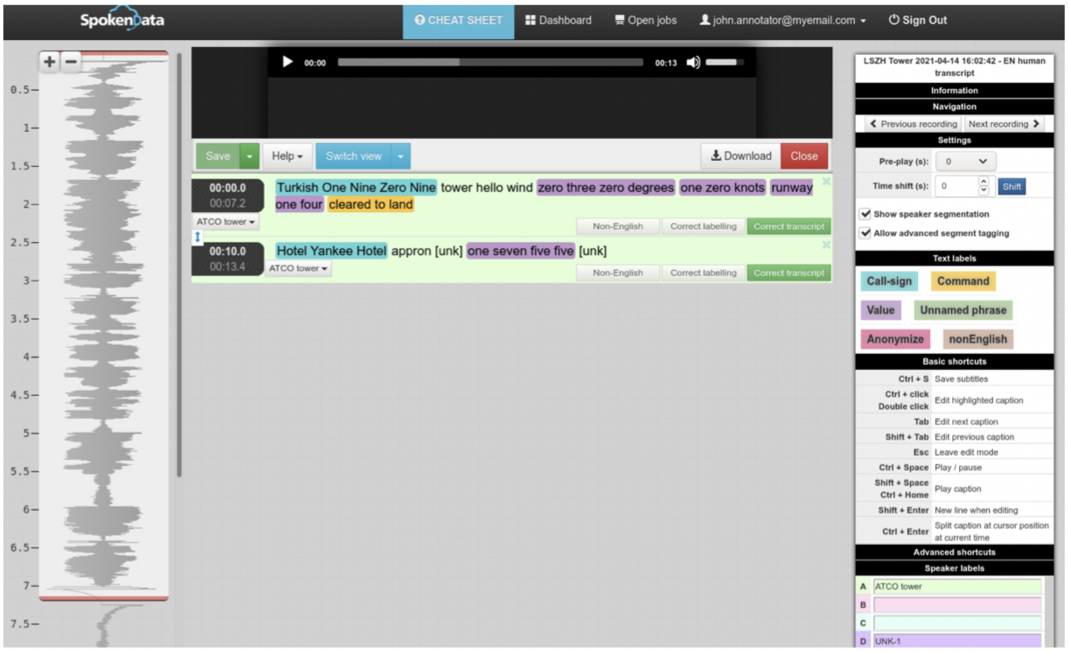
Master thesis already in 2011 [Sho11] showed for the first time that including context knowledge in ASR significantly reduces Word Error Rates (WER) in an ATC task. For instance, the WER was reduced by a factor of almost 10 times i.e., 2.8% to 0.3%. In a follow-up project, AcListant and DLR focused on integrating their ASR into an arrival manager (in order to improve the prediction of the landing sequence). Following MALORCA project focused on ASR directly developed and integrated for two Air-Navigation Service Providers (ANSPs): ANS CR[14] and Austrocontrol[15].

Subsequent (and still ongoing project) HAAWAII focuses on more complex tasks such as read-back error detection while exploring ASR as well. ATCO2 project, which recently ended, developed a unique platform allowing to collect, organize and pre-process ATC (voice communication) data from air space (see Figure 16 for an overview). First the project considered the real-time voice communication between ATCos and pilots available either directly through publicly accessible radio frequency channels, or indirectly from ANSPs. In addition to the voice communication, the contextual information available in a form of metadata (i.e. surveillance data) was exploited, available as part of OpenSky Network services[16].

[14] https://www.ans.cz

[15] https://www.austrocontrol.at

[16] https://opensky-network.org



00:00.0
00:07.2
ATCO tower
Turkish One Nine Zero Nine tower hello wind zero three zero degrees one zero knots runway one four cleared to land

00:10.0
00:13.4
ATCO tower
Hotel Yankee Hotel appron [unk] one seven five five [unk]

LSZH Tower 2021-04-14 16:02:42 - EN human transcript

Pre-play (s): 0
Time shift (s): 0

☑ Show speaker segmentation
☑ Allow advanced segment tagging

Ctrl + S	Save subtitles
Ctrl + click Double click	Edit highlighted caption
Tab	Edit next caption
Shift + Tab	Edit previous caption
Esc	Leave edit mode
Ctrl + Space	Play / pause
Shift + Space Ctrl + Home	Play caption
Shift + Enter	New line when editing
Ctrl + Enter	Split caption at cursor position at current time

A	ATCO tower
B	
C	
D	UNK-1

Figure 16 Automatic transcription and annotation of ATC speech data with possible manual verification.

3 Boosting contextual information in ASR

This section summarises our R&D achievements in an area of boosting contextual information for automatic speech recognition applications. The commented works address all 3 principal components of the ASR systems, namely acoustic modeling, language modeling, and lexical knowledge. Eventually, this section will also give a brief overview of contextual boosting for a particular domain of air-traffic management.

Following 4 papers were selected to be summarised, commented and aligned with other works in this section:

(1) A Comparison of Methods for OOV-Word Recognition on a New Public Dataset, 2021, [BMM21],

(2) English Spoken Term Detection in Multilingual Recordings, 2010, [MVG10],

(3) A Context-Aware Speech recognition and Understanding System for Air Traffic Control Domain, 2017, [Oua+17],

(4) Boosting of contextual information in ASR for air-traffic call-sign recognition, 2021, [Koc+21b].

3.1 Introduction

Contextual boosting - a technique to adapt ASR engine to increase its efficiency towards highly informative content (or phrase) - can be very beneficial for various applications. In practice, the problem of boosting can evolve in many directions. One of them is the recognition of words not seen during training (i.e., often called Out-Of-Vocabulary (OOV) words). Another problem is rather due to a very low occurrence of specific words in training data, thus having a low probability to be recognized due to low n-gram score in language model. This section summarizes our work done in this direction.

Goal of contextual boosting, as briefly introduced above and supported by our past and ongoing work, is to increase the probability of words, or sequence of words, to be recognized with high accuracy by an ASR system. The concept of boosting assumes that some prior information (e.g., list of words to be boosted) is known in advance to the user. The prior information can be similar to the whole test data (e.g. test set) used for recognition, or an online (real-time) ASR can be considered and boosting can then vary for each particular audio file (utterance) to be recognized.

3.2 Boosting of OOV words

(Relevant paper: A COMPARISON OF METHODS FOR OOV-WORD RECOGNITION ON A NEW PUBLIC DATASET, Rudolf Braun, Srikanth Madikeri and Petr Motlicek, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, Ontario, Canada, 2021 [BMM21])

In our recent work from 2021 [BMM21], we addressed the problem of boosting the scores for OOV words by focusing on Weighted Finite-State Transducer (WFST) based ASR systems with distinct acoustic and language models [MPR08]. In these systems both the LM and lexicon are fixed and encoded as a WFST, thus words unseen during training cannot be recognized. Our proposed solution towards this problem was to employ word or subword-based models while using a phone LM as the pronunciation for [unk] token[17], and then try to recover a word from the recognized phone sequence aligned with the [unk] token. A reproducible dataset for English and German using CommonVoice [Ard+19] was built with a large number of realistic OOVs in the test set. Also a new tool for calculating error rate metrics was released[18], and we proposed a new metric called "OOV-CER" for measuring OOV-word recognition performance independent of the performance on in-vocabulary words.

3.3 Boosting by a prior from another modality

(Relevant paper: English Spoken Term Detection in Multilingual Recordings, Petr Motlicek, Fabio Valente and Philip N. Garner, in: Proceedings of Interspeech, Makuhari, Japan, 2010 [MVG10])

Very specific type of boosting can be considered by using a prior extracted from another modality (e.g., a presentation (slides) provided with audio from a conference lecture). Previous section considered incorporation of a prior knowledge (i.e., highly informative words although not included in the ASR lexicon) through a decoding WFST graph. This section proposes to use prior information (available from another modality) through modifying (rescoring) ASR output (word recognition lattices).

The work from 2010 [MVG10] used an English LVCSR based Spoken Term Detection (STD) engine performing automatic indexation of real lecture recordings. The audio recordings were uttered in English (usually by non-native speakers), however, some recordings were partially (e.g. at the beginning of the talk), or fully uttered in French or Italian. Blindly applying an English STD engine for automatically indexing English segments in such multi-lingual recordings would lead to a significant decrease of overall STD performance since the English engine

[17] [unk] (unknown) token is often used in ASR for the words whose pronunciations are represented by a phone LM trained on a lexicon of words with low counts.

[18] https://github.com/idiap/icassp-oov-recognition

would be employed on“inappropriate” speech input (i.e., speech pronounced in different (alien) languages whose words do not appear in the LVCSR dictionary).

One of the solutions would be to to employ a language identification, requiring to encode the knowledge of other (non-target) languages. Another solution is to build an Out-Of-Language (OOL) detection module built around LVCSR word lattices subsequently used for search of the spoken terms.

To perform STD, the recordings are first pre-processed by using the LVCSR system that produces word recognition lattices. The word lattices are then converted into a candidate term index accompanied with times and detection scores. The detection scores are represented by the word posterior probabilities P estimated from the lattices using the forward-backward re-estimation algorithm [EW00], and defined as:

$$P(W_i; t_s, t_e) = \sum_{Q} P(W_i^j; t_s, t_e | x_{t_s}^{t_e}),$$

where W_i is the hypothesized word identity spanning the time interval $t \in (t_s, t_e)$. t_s and t_e denote the start and end time interval, respectively. j denotes the occurrence of word W_i in the lattice. $x_{t_s}^{t_e}$ denotes the corresponding partition of the input speech (the observation feature sequence). Q represents a set of all word hypotheses sequences in the lattice that contain the hypothesized word W_i in $t \in (t_s, t_e)$.

In order to boost some terms which were found apriori in corresponding slides, word posterior probabilities $P(W_i; t_s, t_e)$ of searched terms can be modified by using a prior which represents a relevance of a term to the topic (given by corresponding text slides). The prior is introduced by a multiplicative constant c:

$$\begin{array}{lll} P_{new} = cP_{old}, & if & c <= 1/P_{old}, \\ P_{new} = 1, & & otherwise. \end{array}$$

We tested the boosting algorithm on multi-lingual lecture recordings (supplemented with text slides): (i) for each lecture recording, a new list of terms was automatically generated based on the occurrence of searched terms in the text of corresponding PowerPoint slides. Since no time allocation of the individual slides and their precise alignment with the audio segments of a lecture is available (only the general lecture number assignment), no precise temporal information is employed. (ii) Posterior probabilities P_{old} (initially estimated from the LVCSR based word recognition lattices) associated with search terms occurring in the new list of a given lecture are updated.

Figure 17 graphically shows a dependence of Equal Error Rate (EER)[19] on varying c for two STD systems (without and with application of the OOL detection module). c varied from 10^{-4} to 10^{3}.

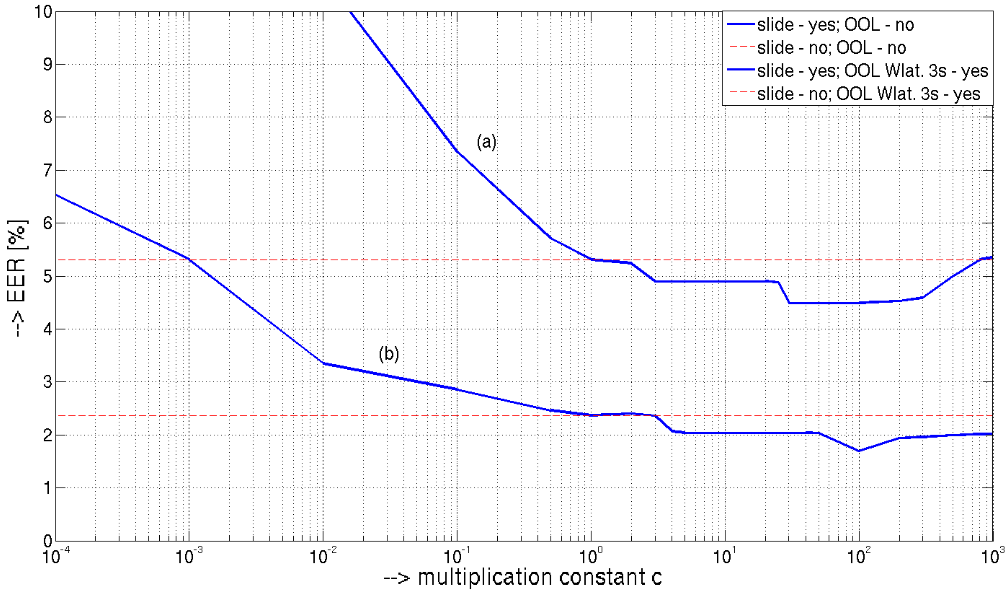


Figure 17 Overall EERs of spoken-term detection when additional prior information is exploited: (a) STD system without OOL module, (b) STD system with OOL module (source: [MVG10]).

3.4 Context-based re-scoring of ASR output

(Relevant paper: A Context-Aware Speech recognition and Understanding System for Air Traffic Control Domain, Youssef Oualil, Dietrich Klakow, Gyorgy Szaszak, Ajay Srinivasamurthy, Hartmut Helmke and Petr Motlicek, in: Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop, Okinawa, Japan, 2017 [Oua+17])

Previous section considered prior information (as textual entities extracted from PowerPoint presentations) to boost term detection in audio streams. This work was done by rescoring the ASR hypotheses.

For some specific domains such as Air-Traffic Control (ATC), the context information can be provided in many different ways. For instance, it can be available as abstract concepts (e.g. airline codes such as “AFR2A”), which are however difficult to map into full possible spoken sentences to perform rescoring (or model adaptation). Our work from 2017 [Oua+17] presented a multi-modal approach, which dynamically integrates partial temporal and situational

[19] The EER is the location on a ROC or DET curve where the false acceptance rate and false rejection rate are equal.

ATC context information to improve its performance. More specifically, we tackled this work also by re-scoring word recognition output (N-best) by using word sequences which carry relevant ATC information as well as by directly adapting a language model (which can be seen as a similar technique to WFST boosting).

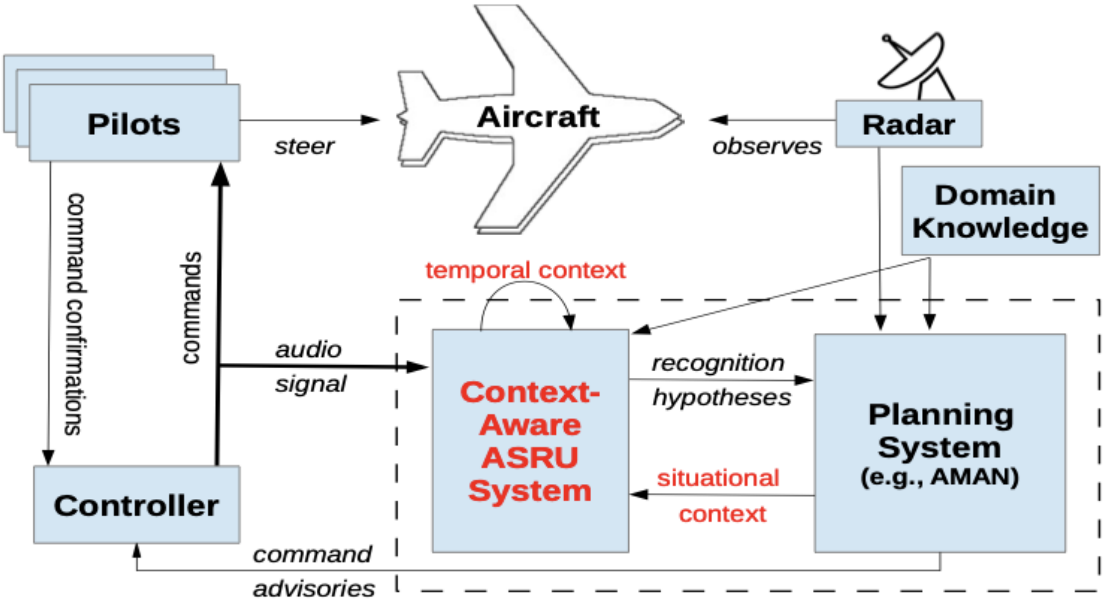


Figure 18 Schematic view of an automatic speech recognition-based ATC system (source: [Oua+17]).

3.4.1 Contextual data in ATC

The ATCos provide the commands to the pilots based on the state of a given airspace sector obtained from radar information. These commands are issued in an irregular way and usually contain an information as: (i) an aircraft call-sign (e.g. **AFR2A** $\sim=$ *air france two alpha*) followed by a command type to execute and a command value to achieve (e.g. **REDUCE 220** $\sim=$ *reduce speed two two zero knots*).

Recognition of ATC commands is a primary goal of ASR based ATC systems. Our solution toward this (already partially described above in Section 2.8) is to incorporate the contextual data (which can be extracted in many ways from radar information available for ANSPs, or from concurrent streams (captured by ADS-B devices)) regularly issued by airplanes[20]. In MALORCA project, we attempted to solve this problem by using the information regularly extracted from the radar, while in more recent project ATCO2, the surveillance data was retrieved directly from the OpenSky Network[21]. As the recorded ATC utterances are stored together with a timestamp,

[20] https://en.wikipedia.org/wiki/Automatic_Dependent_Surveillance\T1\textendashBroadcast
[21] https://opensky-network.org

this timestamp can be used in combination with the ADS-B receiver (or airport) location to send a query to the OpenSky Network (OSN) database. The OSN collects ADS-B and Mode S data from airplanes from many locations around the world. The query to the OSN database has two parameters: the time range and the search area. The time range is centered on the timestamp, and the search area is centered on the receiver (or airport) location. The query returns the ADS-B information from every plane that matches the criteria. The call-signs contained in the ADS-B information are present in the ICAO format (a three-character airline code, e.g., LUF(Lufthansa), followed by the call-sign number, which consists of a digit combination and may also contain an additional character combination, e.g., LUF189AF, this is the compressed form of a call-sign.).

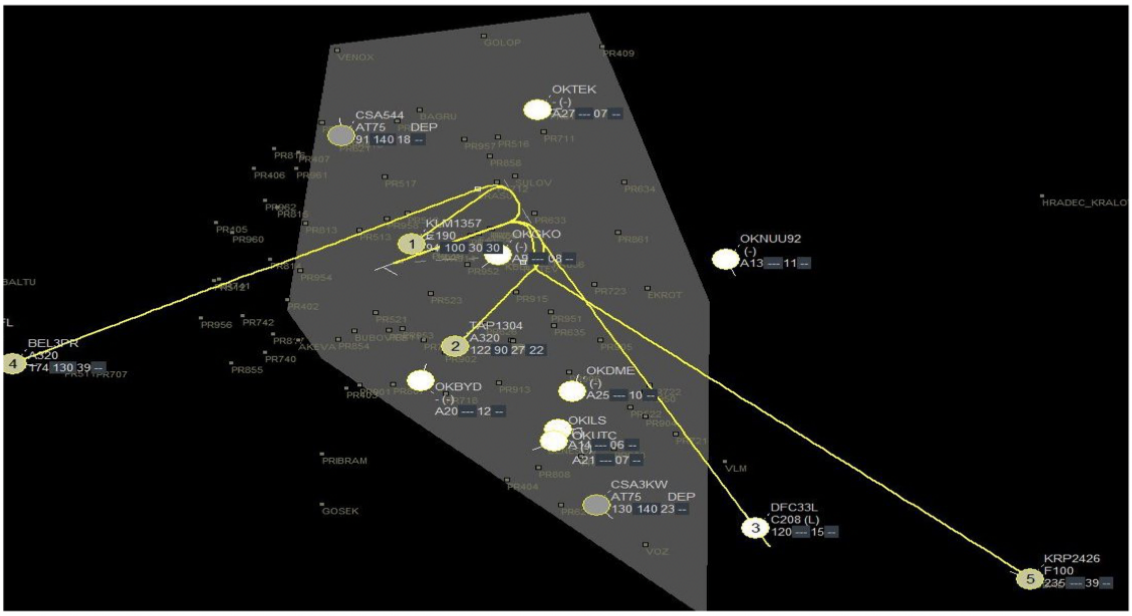
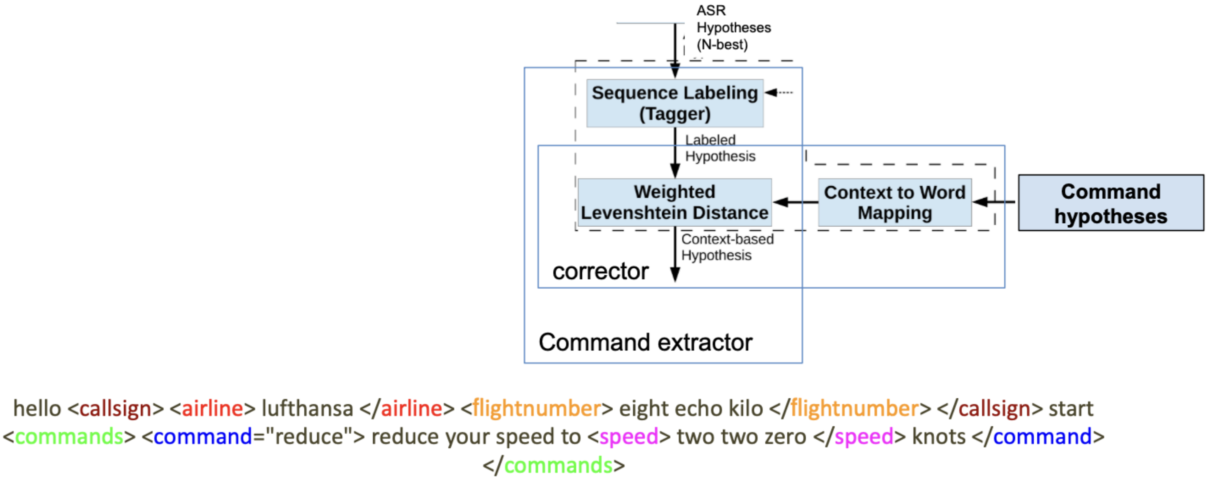


Figure 19 Expected landing sequences and trajectories for different aircraft approaching Prague airport. (source: [Oua+17]).

3.4.2 Context-based rescoring

As part of the 2017 paper [Oua+17], we proposed and developed a rescoring approach which follows these steps (visualised in Figure 20):

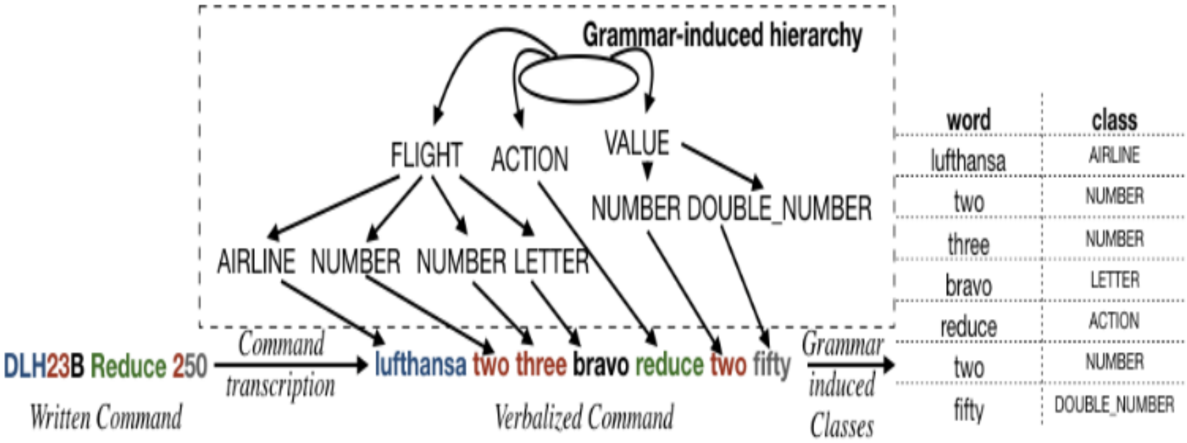
- “Sequence Labeling”: We use a context-free-grammar-based token tagger [Sch+14] (which is developed from rules manually collected by experts) to automatically map the word transcripts provided by ASR to a concept level (Figure 21). As an example – the ASR hypothesis “**air france two alpha hello reduce speed two three zero knots**” is mapped to the following concept “**<callsign> air france two alpha**



hello <callsign> <airline> lufthansa </airline> <flightnumber> eight echo kilo </flightnumber> </callsign> start <commands> <command="reduce"> reduce your speed to <speed> two two zero </speed> knots </command> </commands>

Figure 20 Command extractor and corrector - steps described in Section 3.4.2.

</callsign> hello <command=reduce> reduce speed <speed> two three zero </speed> knots </command>”.



word	class
lufthansa	AIRLINE
two	NUMBER
three	NUMBER
bravo	LETTER
reduce	ACTION
two	NUMBER
fifty	DOUBLE_NUMBER

Figure 21 Use of context-free grammar to transduce ATC segments (text) to concepts.

- “Context-to-Word Mapping”: The partial rescoring approach turns the problem of generating full spoken sentences (realizations) of the context (i.e., from radar) into generating realization of short segments, which can be extracted by the sequence labeler in the previous step. As an example, instead of generating the full realization of the command “**AFR2A REDUCE 250**”, we only need to generate context-to-word mapping for the call-sign “**AFR2A**” and the speed value “**250**”.

- “Context-based Rescoring”: A Weighted Levenshtein Distance (WLD) is used to rescore the segments extracted from the ASR hypotheses (Step 1), to find the closest context segments (i.e., from all verbalized context segments available from Step 2).

3.5 Two-stage boosting

(Relevant paper: Boosting of contextual information in ASR for air-traffic call-sign recognition, Martin Kocour, Karel Vesely, Alexander Blatt, Juan Zuluaga-Gomez, Igor Szoke, Jan Cernocky, Dietrich Klakow and Petr Motlicek, in: Proceedings of Interspeech, 2021 [Koc+21b])

A more advanced algorithm to boost contextual information was developed very recently jointly at BUT and Idiap. It is called a two-stage boosting strategy, consisting of (i) HCLG boosting and (2) lattice boosting, both implemented as WFST compositions. Briefly for HCLG boosting, score discounts are given to individual words, while in lattice boosting the score discounts are given to word-sequences.

The work has so far been developed and tested for boosting the call-signs in ATC applications, nevertheless, the approach is very universal and can be used for boosting words or sequence of words for various scenarios. Specifically, we apply targeted boosting of certain words, or word-strings by applying score discounts into language model scores done by means of WFST composition. The boosted expressions are thus made more likely to appear in the best hypothesis of ASR. This approach is natural for WFST based ASR systems as for instance developed in Kaldi.

3.5.1 List of call-signs

As already described in Section 3.1, prior information (known in advance of boosting) for each spoken utterance (or generic to given “session”) is expected, and can be provided either from radar screen or through a concurrent data stream such as from ADS-B receivers. Concretely, a list of candidate call-signs for given short-term traffic situations can be periodically provided. These call-signs can be obtained in a dynamic way (e.g. from a radar system), in a static way from a historical database of traffic monitoring, or from ADS-B (where the synchronization between speech and ADS-B channels can be done using timestamp and location information).

3.5.2 Lattice boosting

It is done through the composition of L and B, where L means the original lattice, and B means boosting graph. Depending on application, B can be made specific for each speech segment (utterance). Our first implementations aimed at a batch-mode composition (i.e., offline mode), but most recent work (in 04/2022 under submission for Interspeech 2022 conference) describes an implementation in an on-line mode. The composition can be implemented as a fast operation as both the lattices and boosting graphs are relatively small.

3.5.3 HCLG boosting

The HCLG boosting is also done as a composition of two FSTs, $HCLG$ and B, where $HCLG$ is the pre-compiled recognition network[22], and B is another type of boosting graph. The original $HCLG$ graph is plugged into the ASR decoder to generate word-recognition lattices. B can be utterance-specific. The composition of B with $HCLG$ graph is performed on-the-fly immediately before initializing the decoder. An alternative approach to $HCLG$ boosting, already presented in Section 3.2 proposed to boost the G.fst and do on-the-fly composition with HCL.fst graph. This implementation was tested as part of our research report [Nig+21]. The presented paper [Koc+21b] shows that a cascade of HCLG and lattice boosting is complementary and the boosted elements appear more likely as part of the best ASR hypothesis.

3.6 Application of contextual boosting in air-traffic management

(Paper also for consideration in this section: Automatic processing pipeline for collecting and annotating air-traffic voice communication data, Martin Kocour, Karel Vesely, Igor Szoke, Santosh Kesiraju, Juan Zuluaga-Gomez, Alexander Blatt, Amrutha Prasad, Iuliia Nigmatulina, Petr Motlicek, in: Proceedings of 9th OpenSky Symposium, OpenSky Network, Brussels, Belgium, pages 1-9, MDPI, 2020 [Koc+21a])

(Paper also for consideration in this section: Machine Learning of Controller Command Prediction Models from Recorded Radar Data and Controller Speech Utterances, Matthias Kleinert, Hartmut Helmke, Gerald Siol, heiko Ehr, Michael Finke, Youssef Oualil and Ajay Srinivasamurthy, in: Proceedings of the 7th SESAR Innovation Days (SID), University of Belgrade, Belgrade, Serbia, 2017 [Kle+17])

Application of speech processing and automatic speech recognition in air-traffic management has already been introduced in Section 2.8. The topic of boosting plays an essential role to reach low word-error rates (for detection and classification of call-signs as well as for other entities of the ATC communication). In brief, contextual boosting allows to incorporate a prior knowledge known in advance (either from another modality such as radar or ADS-B), or from other sources (e.g., command prediction model [Kle+17]). Although the strategy of boosting was already applied in ASR for ATC in 2017 [Oua+17], substantial improvements were made very recently by implementing several approaches directly within the ASR decoder (e.g., HCLG boosting) or on top of decoder output by boosting directly word-recognition lattices. Also shown in the following Section 4.4, another type of boosting which brought further improvements in recognition accuracies was implemented and integrated as part of the natural language processing stage.

[22] https://kaldi-asr.org/doc/graph.html

4 Natural language understanding on automatically generated textual data

This section summarises our past (including very recent) works falling into the category of natural language understanding, with direct or indirect implications to automatic speech recognition. More specifically, the first part will be related to adapting language models using supervised and unsupervised techniques. This will be followed by a section related to the use of powerful recurrent neural networks in language modeling - directly converted to WFSTs. The last section will be devoted to comment on our very recent work on building a named entity recognizer exploiting prior information extracted from radar data for ATC domain.

Following 3 papers were selected among others to be summarised/commented on and aligned with other works in this section:

- Supervised and unsupervised Web-based language model domain adaptation, 2012 [Lec+12],
- Conversion of Recurrent Neural Network Language Models to Weighted Finite State Transducers for Automatic Speech Recognition, 2012 [LM12],
- A two-step approach to leverage contextual data: speech recognition in air-traffic communications, 2022 [Nig+22].

4.1 Joint ASR and NLP

(Paper also for consideration in this section: Expanded Lattice Embeddings for Spoken Document Retrieval on Informal Meetings, Esaú VILLATORO-TELLO, Srikanth Madikeri, Petr Motlicek, Aravind Ganapathiraju and Alexei V. Ivanov, in: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022 [VIL+22])

Before presenting in more details aforementioned NLP areas (represented by 3 papers), one of the goals of this Section 4 is also to recapitulate and highlight our very recent research directions while addressing downstream applications of automatic speech recognition. As a mature technology, ASR has become an alternative input method in many applications, and is in general considered as an input in several SLU tasks, including Information Retrieval (IR) or more concretely Spoken Document Retrieval (SDR). SDR typically employs ASR transcripts to index and retrieve relevant spoken documents. However, it can be expected that upstream processes (such as ASR) inject errors that would negatively affect the retrieval performance. Our very recent activities aim to jointly address both ASR and NLP (i.e., SDR) systems to eventually improve performance of the whole chain.

One of possible solutions to deal with ASR errors in SDR is to consider multiple alternative hypotheses to augment the input to document retrieval to compensate for the erroneous 1-

best hypothesis. N-best output (i.e., the top “n” scoring hypotheses) is usually available in any of WFST based ASR systems and it can easily be fed to a traditional IR pipeline. One of the problems of n-best is that the concurrent hypotheses are terribly redundant, and do not sufficiently encapsulate the richness of the ASR output usually represented as an acyclic directed graph called the lattice.

Our recent work [VIL+22] (also graphically explained in Figure 22) proposes to utilize the lattice’s constrained minimum path to generate a minimum set of hypotheses that serve as input to the re-ranking phase of IR. The novelty of this approach is the incorporation of the lattice as an input for neural re-ranking by considering a set of hypotheses that represents every arc in the lattice. The obtained hypotheses are encoded through sentence embeddings using BERT-based models, namely SBERT and RoBERTa, and the final ranking of the retrieved segments is obtained with a max-pooling operation over the computed scores among the input query and the hypotheses set. This approach, when tested on a standardised database, presumes that this new set of hypotheses derived from the expanded lattice can significantly improve the SDR performance (when compared with typical n-best ASR output).

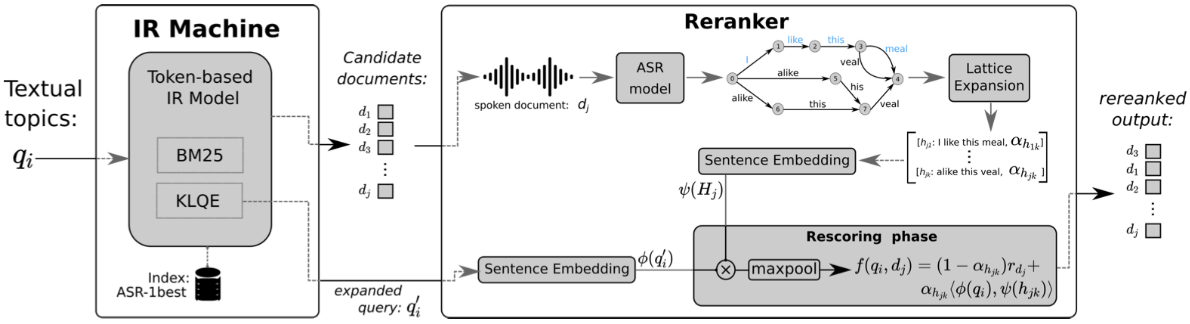


Figure 22 General overview of the proposed multi-stage SDR architecture based on expanded lattice embeddings (source: [VIL+22]).

4.2 Supervised and unsupervised language model adaptation

(Relevant paper: Supervised and unsupervised Web-based language model domain adaptation, Gwénolé Lecorvé, John Dines, Thomas Hain and Petr Motlicek, in: Proceedings of Interspeech, Portland, Oregon, USA, 2012 [Lec+12])

Already partially described in Section 2.4.3, domain adaptation, allowing the use of machine learning technologies (such as ASR) in new environments, is usually required to reach adequate and acceptable performance. In case of LM, the domain adaptation consists in re-estimating probabilities of a baseline (in our case n-gram) LM in order to better match the specifics of a given broad topic of interest. To do so, a common strategy is to retrieve adaptation texts from the Web (e.g. by using Commoncrawl[23]) based on a given domain-representative seed text. In case of our study from 2012 [Lec+12], our goal was to analyze the differences of our Web-based adaptation approach for:

- Supervised case – in which the seed text is manually generated, and
- Unsupervised case – where the seed text is given by an automatic transcript generated by an ASR.

The work was built around video data available at YouTube channels (with manual or automatic transcripts accompanying video).

4.2.1 N-gram LM adaptation

The n-gram LMs are still among the most typical models used by ASR systems and usually require a large multi-topic text collection for training. As a consequence, this LM is not optimal (as part of ASR) to transcribe material dealing with a given specific domain. As proposed solution, LM adaptation (to re-estimate the n-gram probabilities of the baseline LM) can be performed in order to fit the specifics of the considered domain.

Web textual data seems to be a natural alternative to be used for LM adaptation. The process can be split into several steps:

- Query extraction of a text that is representative of the domain of interest – “seed” text. Seed text is of high importance in this process as it is supposed to well characterize the target domain so that we can extract meaningful information (documents) from the Web.
- Retrieve web pages by submitting the queries to a Web search engine.
- Build an adapted LM by integrating the retrieved adaptation data with background training material.

Having a large amount of seed text is desirable as it can better represent the domain of interest. However in case of domain such as those related to spontaneous speech (e.g., multiparty meetings[24], etc.), this might be problematic as such data do not really exist/cannot be easily retrieved from Web, or it is costly to produce such text by manually transcribing the spoken material (video/audio). Therefore, supervised LM adaptation is not always feasible. In the case of automatizing this process and relying more on automatically generated text data, this would lead to a much lower effort by humans/developers.

[23] https://commoncrawl.org

[24] https://www.cstr.ed.ac.uk/research/projects/ami

Overall, the work commented in this section compared the Web-based domain LM adaptation process using different levels of supervision while also analysing the impact of recognition errors in the seed text on ASR accuracy gains provided by LM adaptation and the dependence on the size of the seed text. Achieved results indicate that the use of manual transcripts brings the greatest improvement in terms of perplexity and ASR accuracy. Further we also found out that the recognition errors do not significantly bias LM adaptation, as this is usually the case for query extraction, or for linear interpolation. This is very interesting due to the fact that error spotting in ASR outputs is a complex task. Finally, the presented work has demonstrated that reducing the size of the seed text does not change aforementioned observations. In fact, results indicate that decreasing the seed text size reduces both the gains in perplexity and in word error rates consistently for both supervised and unsupervised cases, though in the unsupervised case this is more pronounced.

4.3 Conversion of RNN based LM to WFST

(Relevant paper: Conversion of Recurrent Neural Network Language Models to Weighted Finite State Transducers for Automatic Speech Recognition, Gwénolé Lecorvé and Petr Motlicek, in: Proceedings of Interspeech, Portland, Oregon, USA, 2012 [LM12])

Recurrent Neural Network Language Models (RNNLMs) have been known to significantly increase accuracies of ASR when used on top of n-gram LM. In fact, highly cited paper of colleagues from BUT in 2010 have presented the work on the use of RNNLMs in ASR and demonstrated up to 50% reduction of perplexity by using mixture of several RNN LMs [Mik+10].

Our work on the use of RNNs in LM was already briefly introduced in Section 2.3. Specifically, RNNLMs are used in a two-pass ASR approach to re-score N-best lists generated in the first-pass by using n-gram LMs. This means that the prediction power of RNNLMs is used only on subsets of all transcription hypotheses. which implies that the approach does not offer the optimal solution since the n-gram LM used for the first-pass (decoding) may have discarded hypotheses which the RNNLM would have judged very likely. It has also been shown that both n-gram and RNNs provide complementary distributions [Mik+11] and thus the use of RNNLMs in early stages of speech decoding is a challenging objective.

The problem of employing RNNLM in the first-pass ASR has been addressed by our paper in 2012 [LM12]. The key obstacle for ASR is that RNNLMs cannot be used to directly decode a speech signal since they rely on continuous representations of word histories while decoding algorithms (e.g. the one implemented in Kaldi [Pov+11]) require to handle discrete representations to remain tractable. In our work, we defined a new generic strategy to transform RNNLMs into a Weighted Finite State Transducer (WFST) which can directly be used within the decoding process in ASR. The principle of the conversion consists in discretizing continuous RNNLM representations of word histories in order to build WFST states, and then to link these states with probabilities derived from the RNNLM. Figure 23 gives a graphical overview of the discretization scheme. In practice, this approach also raises some needs for pruning the generated WFST since the theoretical number of states may be large according to the chosen discretization strategy. The paper presented a preliminary implementation of the RNNLM conversion algorithm based on K-means clustering and entropy pruning.

Although the obtained results brought only marginal improvements (i.e., compared to the approach where RNNLMs were employed to re-score N-best hypotheses in the second pass of ASR), the approach is theoretically valid and the problem is still (even nowadays) interesting from a research point of view.

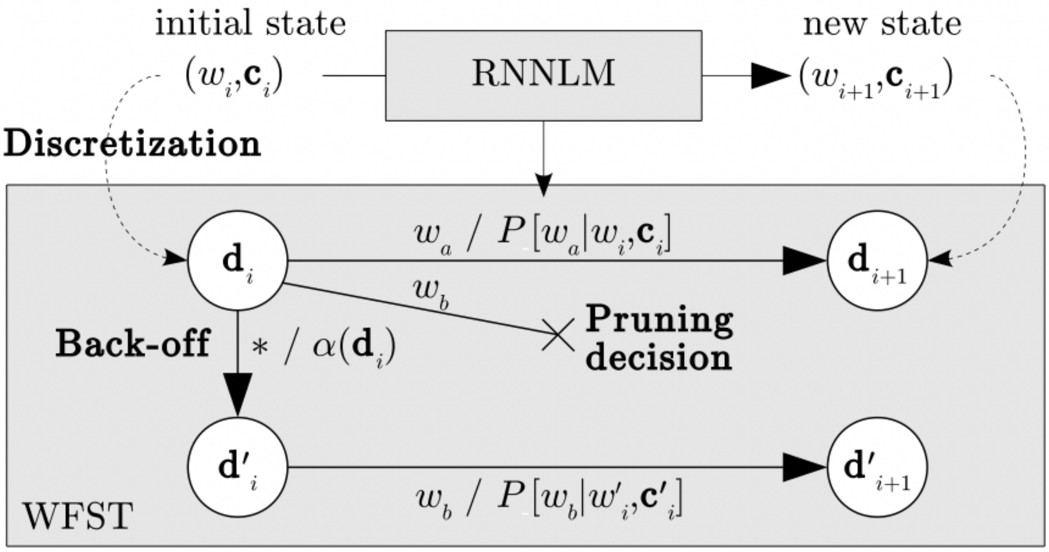


Figure 23 Overview of the RNNLM discretization scheme.

4.4 Boosting of NER for air-traffic management

(Relevant paper: A two-step approach to leverage contextual data: speech recognition in air-traffic communications, Iuliia Nigmatulina, Juan Zuluaga-Gomez, Amrutha Prasad, Seyyed Saeed Sarfjoo and Petr Motlicek, in: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2022 [Nig+22])

In addition to the work presented above in Section 4.1, this section presents the work on a transition between both ASR and NLP technologies. The work, purely dedicated to the ATM domain, is an extension of the previous work already presented in Section 3.5 on (two stage) contextual boosting. Unlike previous work presented in [Koc+21b], the work briefly presented here aims to combine the benefits of ASR and NLP methods and to demonstrate that the use of surveillance data (i.e. available through additional modality) applied simultaneously in both technologies helps to considerably improve the ASR (i.e., recognition of call-signs considered as named entities). Practically this work can be seen as a two-step call-sign boosting approach, where:

- at step (1) as part of ASR - weights of probable call-sign n-grams are reduced (i.e., boosted) in G.fst and/or in the decoding FST (lattices), and
- at step (2) as part of NLP - call-signs extracted from the boosted recognition outputs by using Named Entity Recognition (NER) module are eventually correlated with the surveillance data to select the most suitable one.

Our work demonstrates that ASR and NLP can be seen as complementary tasks rather than separated ones. Whereas ASR exploits speech to produce a sequence of words, NLP exploits the intrinsic characteristics in a given snippet of text. ASR normally struggles to model long sequences, while state-of-the-art NLP systems allow extracting key information in the whole chunks of text; for instance an entire ATC utterance. The proposed approach focuses on an iterative use of contextual data to take advantage of a combination of ASR and NLP modules.

The ASR engine does not differ much from the one described in [Koc+21b]. The NER module is built using BERT model [Dev+18], pre-trained as masked language model from Huggingface [Wol+19] and fine-tuned it on NER task with our in-domain text (i.e., where each word has a tag). A data augmentation pipeline was also implemented in order to increase the amount of training data. The developed NER is then capable of extracting the call-sign information from a given transcript, or in our case from ASR 1-best hypotheses. Recognition of the call-sign entity is crucial where a single error produced by the ASR system affects the whole entity (normally composed of three to eight words).

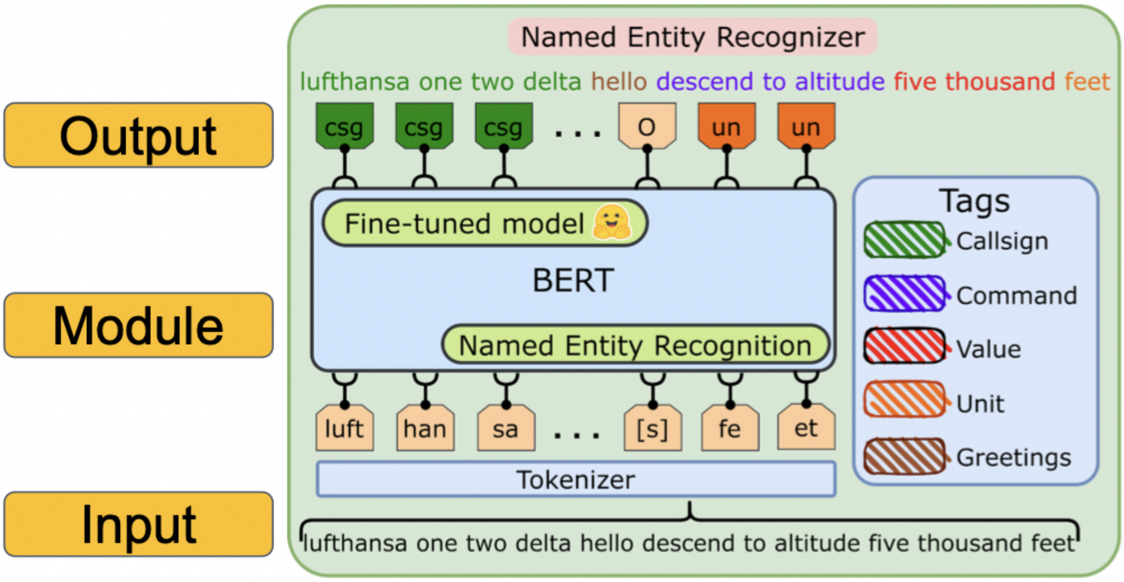


Figure 24 Call-sign and command tagging: Named entity recognizer to automatically tag the input word sequence for respective ATC classes. Visualisation of BERT-based model (Huggingface) fine-tuned on NER task.

4.4.1 Re-ranking

The output of an NER system is a list of tags that match words or sequences of words in an input utterance. As our only available source of contextual knowledge (provided by radar - see Figure 25) in this work are call-signs registered at a certain time and location, we extracted call-signs with the NER system and discarded other entities. Correspondingly, each utterance has a list of call-signs expanded into word sequences. As input, the re-ranking module takes (i) a call-sign extracted by the NER system and (ii) an expanded list of call-signs (available from radar). The re-ranking module compares a given n-gram sequence against a list of possible n-grams, and finds the closest match from the list of surveillance data based on the weighted Levenshtein distance.

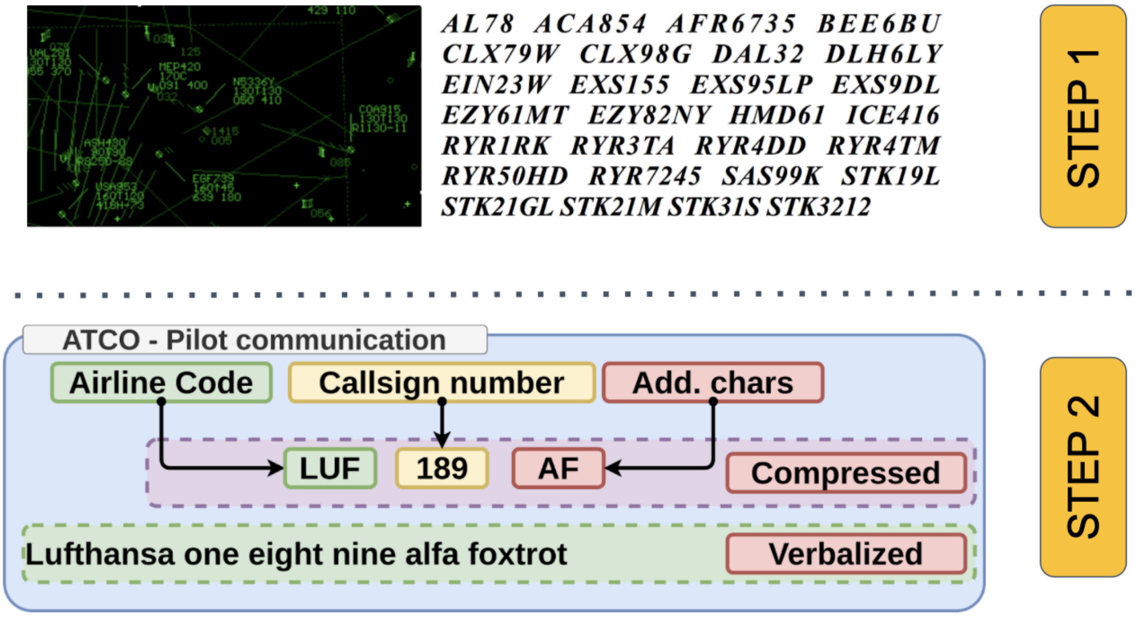
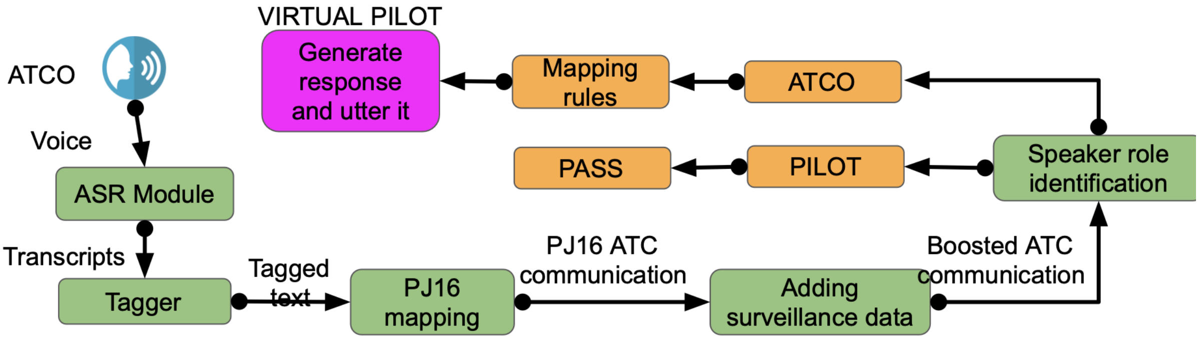

AL78 ACA854 AFR6735 BEE6BU
CLX79W CLX98G DAL32 DLH6LY
EIN23W EXS155 EXS95LP EXS9DL
EZY61MT EZY82NY HMD61 ICE416
RYR1RK RYR3TA RYR4DD RYR4TM
RYR50HD RYR7245 SAS99K STK19L
STK21GL STK21M STK31S STK3212
STEP 1
ATCO - Pilot communication
Airline Code
Callsign number
Add. chars
LUF
189
AF
Compressed
Lufthansa one eight nine alfa foxtrot
Verbalized
STEP 2

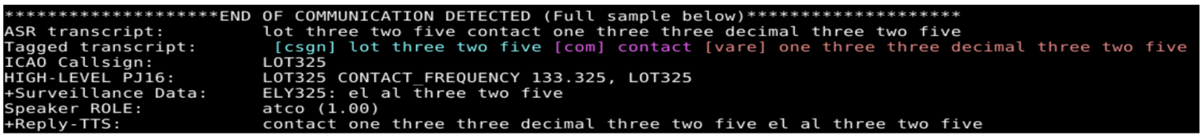
Figure 25 Integrate the surveillance data (extracted from radar screen) into the ASR system.

4.4.2 BERT - as speaker role detector

As a subsequent task in the ATM domain, also implying the use of the BERT-based model, is represented by a speaker role detector. Similarly to the NER module (developed from BERT), we also trained the speaker role detector as a text-based module to reliably classify the utterances into two classes: Air-Traffic Controllers (ATCos), or pilots. Implementation of this module on purely acoustics, e.g., by expecting that the SNR level of pilot's speech will be way lower than SNR of ATCos, was not found very reliable. Although not anticipated, the communication channel between air-traffic controllers and pilots is often not automatically split (i.e., it is available as a 1-channel) and thus automatic and reliable speaker segmentation is of high interest - as a pre-processing block for all downstream applications. A recent work on speaker role detection (while using a rule-based approach) can be found in the research report [Pra+21].

Overview of the whole processing scheme including ASR module to automatically transcribe the ATCo-pilot voice communication, followed by NER tagging and speaker role identification, can be seen in Figure 26. The application is assumed to work as a virtual pilot, automatically answering to the controllers on the issued commands.





```
********************END OF COMMUNICATION DETECTED (Full sample below)********************
ASR transcript:         lot three two five contact one three three decimal three two five
Tagged transcript:       [csgn] lot three two five [com] contact [vare] one three three decimal three two five
ICAO Callsign:          LOT325
HIGH-LEVEL PJ16:        LOT325 CONTACT_FREQUENCY 133.325, LOT325
+Surveillance Data:     ELY325: el al three two five
Speaker ROLE:           atco (1.00)
+Reply-TTS:             contact one three three decimal three two five el al three two five
```

Figure 26 Overview of an ATM application of virtual pilot: besides ASR, the module for speaker role identification and text-to-speech (to generate an automatic response back to pilot) are deployed.

References

[ABM08] Guillermo Aradilla, Herve Bourlard, and Mathew Magimai-Doss. “Using KL-based Acoustic Models in a Large Vocabulary Recognition Task”. In: *Proc. of Interspeech* (Jan. 2008).

[AVB07] Guillermo Aradilla, Jithendra Vepa, and Hervé Bourlard. “An Acoustic Model Based on Kullback-Leibler Divergence for Posterior Features”. In: *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*. IDIAP-RR 06-60. 2007.

[Ard+19] Rosana Ardila et al. “Common Voice: A Massively-Multilingual Speech Corpus”. In: *CoRR* abs/1912.06670 (2019). arXiv: 1912.06670. URL: http://arxiv.org/abs/1912.06670.

[Bae+20] Alexei Baevski et al. *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*. 2020. DOI: 10.48550/ARXIV.2006.11477. URL: https://arxiv.org/abs/2006.11477.

[Bah+86] L. R. Bahl et al. “Maximum mutual information estimation of hidden Markov model parameters for speech recognition”. In: *in Proc. IEEE ICASSP*. Vol. 1. Apr. 1986, pp. 49–52.

[BM94] H. Bourlard and N. Morgan. *Connectionist Speech Recognition – A Hybrid Approach*. Kluwer Academic Publishers, 1994.

[Bou+11] Hervé Bourlard et al. “Current trends in multilingual speech processing”. In: *Sadhana* 36.5 (Oct. 2011), pp. 885–915. DOI: 10.1007/s12046-011-0050-4. URL: http://www.ias.ac.in/sadhana/Pdf2011Oct/885.pdf.

[BMM21] Rudolf Braun, Srikanth Madikeri, and Petr Motlicek. “A COMPARISON OF METHODS FOR OOV-WORD RECOGNITION ON A NEW PUBLIC DATASET”. In: *2021 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE Signal Processing Society. Toronto, Ontario, Canada, June 2021.

[BD91] John S. Bridle and L. Dodd. “An Alphanet approach to optimising input transformations for continuous speech recognition”. In: *1991 International Conference on Acoustics, Speech, and Signal Processing, ICASSP ’91, Toronto, Ontario, Canada, May 14-17, 1991*. IEEE Computer Society, 1991, pp. 277–280. DOI: 10.1109/ICASSP.1991.150331. URL: https://doi.org/10.1109/ICASSP.1991.150331.

[Bur+10] Lukáš Burget et al. “Multilingual acoustic modeling for speech recognition based on subspace Gaussian Mixture Models”. In: *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2010, pp. 4334–4337. DOI: 10.1109/ICASSP.2010.5495646.

[Con+20] Alexis Conneau et al. *Unsupervised Cross-lingual Representation Learning for Speech Recognition*. 2020. DOI: 10.48550/ARXIV.2006.13979. URL: https://arxiv.org/abs/2006.13979.

[DM80] Steven Davis and Paul Mermelstein. “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences”. In: *IEEE transactions on acoustics, speech, and signal processing* 28.4 (1980), pp. 357–366.

[DHK13] Li Deng, Geoffrey Hinton, and Brian Kingsbury. “New types of deep neural network learning for speech recognition and related applications: an overview”. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2013, pp. 8599–8603. DOI: 10.1109/ICASSP.2013.6639344.

[Dev+18] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2018. DOI: 10.48550/ARXIV.1810.04805. URL: https://arxiv.org/abs/1810.04805.

[Eur12] Eurocontrol. *LINK2000+: ATC data link operational guidance in support of DLS regulation*. 2012. URL: https://www.skybrary.aero/bookshelf/books/2383.pdf.

[EW00] G. Evermann and P.C. Woodland. "Large vocabulary decoding and confidence estimation using word posterior probabilities". In: *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*. Vol. 3. 2000, 1655–1658 vol.3. DOI: 10.1109/ICASSP.2000.862067.

[Gra+06] Alex Graves et al. "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural 'networks". In: vol. 2006. Jan. 2006, pp. 369–376. DOI: 10.1145/1143844.1143891.

[Had+18] Hossein Hadian et al. "End-to-end Speech Recognition Using Lattice-free MMI". In: Sept. 2018, pp. 12–16. DOI: 10.21437/Interspeech.2018-1423.

[HMO18] Weipeng He, Petr Motlicek, and Jean-Marc Odobez. "Deep Neural Networks for Multiple Speaker Detection and Localization". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. Brisbane, AUSTRALIA, May 2018, pp. 74–79. ISBN: 978-1-5386-3081-5. DOI: 10.1109/ICRA.2018.8461267.

[Hel+17] Hartmut Helmke et al. "Increasing ATM efficiency with assistant based speech recognition". In: *Proc. of the 13th USA/Europe Air Traffic Management Research and Development Seminar, Seattle, USA*. 2017.

[Hel+16] Hartmut Helmke et al. "Reducing controller workload with automatic speech recognition". In: *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*. IEEE. 2016, pp. 1–10.

[Her89] Hynek Hermansky. "Perceptual linear predictive (PLP) analysis of speech". In: (Nov. 1989).

[HES00] Hynek Hermansky, D.P.W. Ellis, and Shilpa Sharma. "Tandem connectionist feature extraction for conventional HMMsystems". In: vol. 3. Feb. 2000, 1635–1638 vol.3. ISBN: 0-7803-6293-4. DOI: 10.1109/ICASSP.2000.862024.

[Hol+15] Harald Holone et al. "Possibilities, challenges and the state of the art of automatic speech recognition in air traffic control". In: *International Journal of Computer and Information Engineering* 9.8 (2015), pp. 1940–1949.

[Hsu07] Bo-June Hsu. "Generalized linear interpolation of language models". In: *2007 IEEE Workshop on Automatic Speech Recognition Understanding (ASRU)*. 2007, pp. 136–140. DOI: 10.1109/ASRU.2007.4430098.

[Ims+12] David Imseng et al. "Comparing different acoustic modeling techniques for multilingual boosting". In: *Proceedings of Interspeech*. Portland, Oregon, Sept. 2012.

[Ims+13a] David Imseng et al. "Impact of deep MLP architecture on different acoustic modeling techniques for under-resourced speech recognition". In: *Proceedings of the IEEE workshop on Automatic Speech Recognition and Understanding*. Dec. 2013.

[Ims+13b] David Imseng et al. "Using out-of-language data to improve an under-resourced speech recognizer". In: *Speech Communication* (2013). ISSN: 0167-6393. DOI: 10.1016/j.specom.2013.01.007. URL: http://www.sciencedirect.com/science/article/pii/S0167639313000101.

[Jel80] Frederick Jelinek. "Interpolated estimation of Markov source parameters from sparse data". In: 1980.

[Jel91] Frederick Jelinek. “Up from trigrams! - the struggle for improved language models”. In: *EUROSPEECH*. 1991.

[Kho+20] Banriskhem Khonglah et al. “INCREMENTAL SEMI-SUPERVISED LEARNING FOR MULTI-GENRE SPEECH RECOGNITION”. In: *Proceedings of ICASSP 2020*. 2020.

[Kho+19] Banriskhem Khonglah et al. *STACKED NEURAL NETWORKS WITH PARAMETER SHARING FOR MULTILINGUAL LANGUAGE MODELING*. Idiap-RR Idiap-RR-12-2019. Idiap, Oct. 2019.

[Kle+17] Matthias Kleinert et al. “Machine Learning of Controller Command Prediction Models from Recorded Radar Data and Controller Speech Utterances”. In: *Proceedings of the 7th SESAR Innovation Days (SID)*. University of Belgrade. Belgrade, Serbia, Nov. 2017.

[Koc+21a] Martin Kocour et al. “Automatic processing pipeline for collecting and annotating air-traffic voice communication data”. In: *Proceedings of 9th OpenSky Symposium 2020*. OpenSky Network. Brussels, Belgium: MDPI, Nov. 2021, pp. 1–9.

[Koc+21b] Martin Kocour et al. “Boosting of contextual information in ASR for air-traffic call-sign recognition”. In: *Interspeech 2021*. Aug. 2021.

[Kub+94] F. Kubala et al. “The Hub and Spoke Paradigm for CSR Evaluation”. In: *Human Language Technology Conference: Proceedings of the workshop on Human Language Technology*. Plainsboro, NJ, 1994, pp. 37–42.

[KL51] S. Kullback and R. A. Leibler. “On Information and Sufficiency”. In: *The Annals of Mathematical Statistics* 22.1 (1951), pp. 79–86. ISSN: 00034851. URL: http://www.jstor.org/stable/2236703.

[LM12] Gwénolé Lecorvé and Petr Motlicek. “Conversion of Recurrent Neural Network Language Models to Weighted Finite State Transducers for Automatic Speech Recognition”. In: *Proceedings of Interspeech*. Portland, Oregon, USA, Sept. 2012, to appear.

[Lec+12] Gwénolé Lecorvé et al. “Supervised and unsupervised Web-based language model domain adaptation”. In: *Proceedings of Interspeech*. Portland, Oregon, USA, Sept. 2012, pp. 131–134.

[Lin+09] Hui Lin et al. “A study on multilingual acoustic modeling for large vocabulary ASR”. In: *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2009, pp. 4333–4336. DOI: 10.1109/ICASSP.2009.4960588.

[MMB21] Srikanth Madikeri, Petr Motlicek, and Hervé Bourlard. “Multitask adaptation with Lattice-Free MMI for multi-genre speech recognition of low resource languages”. In: *Proceedings of Interspeech 2021*. 2021.

[Mad+20a] Srikanth Madikeri et al. “Lattice-Free Maximum Mutual Information Training of Multilingual Speech Recognition System”. In: *In Proceedings of Interspeech 2020*. ISCA, 2020, pp. 4746–4750.

[Mad+20b] Srikanth Madikeri et al. *Pkwrap: a PyTorch Package for LF-MMI Training of Acoustic Models*. 2020. DOI: 10.48550/ARXIV.2010.03466. URL: https://arxiv.org/abs/2010.03466.

[MMG07] Hari Krishna Maganti, Petr Motlicek, and Daniel Gatica-Perez. “Unsupervised Speech/Non-speech Detection for Automatic Speech Recognition in Meeting Rooms”. In: *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*. IDIAP-RR 06-57. 2007.

[Man11] Christian Mandery. "Distributed N-Gram Language Models : Application of Large Models to Automatic Speech Recognition". In: *Computer Science*. 2011.

[Mik+11] Tomas Mikolov et al. "Empirical Evaluation and Combination of Advanced Language Modeling Techniques." In: *INTERSPEECH*. ISCA, 2011, pp. 605–608. URL: http://dblp.uni-trier.de/db/conf/interspeech/interspeech2011.html#MikolovDKBC11.

[Mik+10] Tomas Mikolov et al. "Recurrent neural network based language model". In: vol. 2. Jan. 2010, pp. 1045–1048.

[MPR08] Mehryar Mohri, Fernando Pereira, and Michael Riley. "Speech Recognition with Weighted Finite-State Transducers". In: *Springer Handbook of Speech Processing*. Ed. by Jacob Benesty, M. Mohan Sondhi, and Yiteng Arden Huang. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 559–584. ISBN: 978-3-540-49127-9. DOI: 10.1007/978-3-540-49127-9_28. URL: https://doi.org/10.1007/978-3-540-49127-9_28.

[MVG10] Petr Motlicek, Fabio Valente, and Philip N. Garner. "English Spoken Term Detection in Multilingual Recordings". In: *Proceedings of Interspeech, Makuhari, Japan, 2010*. ISCA. Makuhari, Japan, Sept. 2010.

[MVS12] Petr Motlicek, Fabio Valente, and Igor Szoke. "IMPROVING ACOUSTIC BASED KEYWORD SPOTTING USING LVCSR LATTICES". In: *Proceedings on IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. Japan, Mar. 2012, pp. 4413–4416.

[Mot+15] Petr Motlicek et al. "Exploiting foreign resources for DNN-based ASR". In: *EURASIP Journal on Audio, Speech, and Music Processing* 2015:17 (June 2015). DOI: 10.1186/s13636-015-0058-5.

[NEK94] Hermann Ney, Ute Essen, and Reinhard Kneser. "On structuring probabilistic dependences in stochastic language modelling". In: *Computer Speech Language* 8.1 (1994), pp. 1–38. ISSN: 0885-2308. DOI: https://doi.org/10.1006/csla.1994.1001. URL: https://www.sciencedirect.com/science/article/pii/S0885230884710011.

[Nig+22] Iuliia Nigmatulina et al. "A two-step approach to leverage contextual data: speech recognition in air-traffic communications". In: *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. 2022.

[Nig+21] Iuliia Nigmatulina et al. *Improving callsign recognition with air-surveillance data in air-traffic communication*. Idiap-RR Idiap-RR-20-2021. Idiap, Nov. 2021. URL: https://arxiv.org/abs/2108.12156.

[Oua+17] Youssef Oualil et al. "A Context-Aware Speech recognition and Understanding System for Air Traffic Control Domain". In: *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*. Okinawa, Japan, Dec. 2017.

[Oua+15] Youssef Oualil et al. "Real-time integration of dynamic context information for improving automatic speech recognition". In: *Sixteenth Annual Conference of the International Speech Communication Association*. 2015.

[Pan+15] Vassil Panayotov et al. "Librispeech: An ASR corpus based on public domain audio books". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015, pp. 5206–5210. DOI: 10.1109/ICASSP.2015.7178964.

[PM19] Shantipriya Parida and Petr Motlicek. “Abstract Text Summarization: A Low Resource Challenge”. In: *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2019)*. HongKong, China: Association for Computational Linguistics (ACL), Nov. 2019, p. 5.

[Pas+19] Adam Paszke et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. 2019. DOI: 10.48550/ARXIV.1912.01703. URL: https://arxiv.org/abs/1912.01703.

[PB92] D. Paul and J. Baker. “The Design for the Wall Street Journal-based CSR Corpus”. In: *Human Language Technology Conference: Proceedings of the Workshop on Speech and Natural Language*. Harriman, NY, 1992, pp. 357–362.

[Pov+08] Daniel Povey et al. “Boosted MMI for model and feature-space discriminative training.” In: *ICASSP*. IEEE, 2008, pp. 4057–4060. ISBN: 1-4244-1484-9. URL: http://dblp.uni-trier.de/db/conf/icassp/icassp2008.html#PoveyKKRSV08.

[Pov+10] Daniel Povey et al. “Subspace Gaussian Mixture Models for speech recognition”. In: Mar. 2010, pp. 4330–4333. DOI: 10.1109/ICASSP.2010.5495662.

[Pov+11] Daniel Povey et al. “The Kaldi Speech Recognition Toolkit”. In: *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Catalog No.: CFP11SRW-USB. Hilton Waikoloa Village, Big Island, Hawaii, US: IEEE Signal Processing Society, Dec. 2011.

[Pra+21] Amrutha Prasad et al. *Grammar Based Identification Of Speaker Role For Improving ATCO And Pilot ASR*. Idiap-RR Idiap-RR-22-2021. Idiap, Dec. 2021.

[Rab89] L. R. Rabiner. “A tutorial on hidden Markov models and selected applications in speech recognition”. In: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286.

[Rag+16] Anton Ragni et al. “Multi-Language Neural Network Language Models”. In: Sept. 2016, pp. 3042–3046. DOI: 10.21437/Interspeech.2016-371.

[Rud17] Sebastian Ruder. “An Overview of Multi-Task Learning in Deep Neural Networks”. In: *CoRR* abs/1706.05098 (2017). arXiv: 1706.05098. URL: http://arxiv.org/abs/1706.05098.

[SMM21] Seyyed Saeed Sarfjoo, Srikanth Madikeri, and Petr Motlicek. “Speech Activity Detection Based on Multilingual Speech Recognition System”. In: *Interspeech*. 2021.

[Sch+14] Anna Schmidt et al. “Context-based recognition network adaptation for improving on-line ASR in Air Traffic Control”. In: *2014 IEEE Spoken Language Technology Workshop (SLT)*. 2014, pp. 13–18. DOI: 10.1109/SLT.2014.7078542.

[Sho11] Todd Shore. “Knowledge-based word lattice re-scoring in a dynamic context”. MA thesis. Saarland University, 2011.

[Sri+17] Ajay Srinivasamurthy et al. “Semi-supervised Learning with Semantic Knowledge Extraction for Improved Speech Recognition in Air Traffic Control”. In: *Proceedings of Interspeech 2017*. Stockholm, Sweden, Aug. 2017, pp. 2406–2410. DOI: http://dx.doi.org/10.21437/Interspeech.2017-1446.

[Sto02] Andreas Stolcke. “SRILM - an extensible language modeling toolkit.” In: *INTERSPEECH*. Ed. by John H. L. Hansen and Bryan L. Pellom. ISCA, 2002. URL: http://dblp.uni-trier.de/db/conf/interspeech/interspeech2002.html#Stolcke02.

[TGB17] Sibo Tong, Philip N. Garner, and Hervé Bourlard. “An Investigation of Deep Neural Networks for Multilingual Speech Recognition Training and Adaptation”. In: *Proc. of Interspeech*. Aug. 2017.

[TGB19] Sibo Tong, Philip N. Garner, and Hervé Bourlard. "AN INVESTIGATION OF MULTILINGUAL ASR USING END-TO-END LF-MMI". In: *International Conference on Acoustics, Speech and Signal Processing*. 2019.

[VHB13] Karel Veselý, Mirko Hannemann, and Lukáš Burget. "Semi-supervised training of Deep Neural Networks". In: *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. 2013, pp. 267–272. DOI: 10.1109/ASRU.2013.6707741.

[VMC19] Thibault Viglino, Petr Motlicek, and Milos Cernak. "End-to-End Accented Speech Recognition". In: *Proc. Interspeech 2019*. 2019, pp. 2140–2144. DOI: 10.21437/Interspeech.2019-2122.

[VIL+22] Esaú VILLATORO-TELLO et al. *Expanded Lattice Embeddings for Spoken Document Retrieval on Informal Meetings*. Idiap-RR Idiap-RR-06-2022. Accepted as short papaer at SIGIR 2022. Idiap, Apr. 2022.

[Vu+14] Ngoc Thang Vu et al. "Multilingual Deep Neural Network based Acoustic Modeling For Rapid Language Adaptation". In: *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*. Florence: IEEE, May 2014, pp. 7639–7643. DOI: 10.1109/ICASSP.2014.6855086.

[VMB21] Apoorv Vyas, Srikanth Madikeri, and Hervé Bourlard. "Comparing CTC and LFMMI for out-of-domain adaptation of wav2vec 2.0 acoustic model". In: *Proceedings of Interspeech*. Sept. 2021. URL: https://arxiv.org/abs/2104.02558.

[Wol+19] Thomas Wolf et al. *HuggingFace's Transformers: State-of-the-art Natural Language Processing*. 2019. DOI: 10.48550/ARXIV.1910.03771. URL: https://arxiv.org/abs/1910.03771.

[WP00] Philip Woodland and Daniel Povey. "Large Scale Discriminative Training For Speech Recognition". In: (Dec. 2000).

[Zul+20a] Juan Zuluaga-Gomez et al. "Automatic Call Sign Detection: Matching Air Surveillance Data with Air Traffic Spoken Communications". In: *Proceedings of 8th OpenSky Symposium 2020*. Vol. 59. 1 14. OpenSky Network. MDPI, Nov. 2020, pp. 1–10. DOI: https://doi.org/10.3390/proceedings2020059014. URL: https://www.mdpi.com/2504-3900/59/1/14.

[Zul+20b] Juan Zuluaga-Gomez et al. "Automatic Speech Recognition Benchmark for Air-Traffic Communications". In: *Proc. Interspeech 2020*. Oct. 2020, pp. 2297–2301. DOI: 10.21437/Interspeech.2020-2173.

[Zul+21] Juan Zuluaga-Gomez et al. "Contextual Semi-Supervised Learning: An Approach To Leverage Air-Surveillance and Untranscribed ATC Data in ASR Systems". In: *Interspeech 2021*. Aug. 2021. URL: https://arxiv.org/abs/2104.03643.

5 Selected papers underlying this thesis

Acknowledgment of the publishers. Due to copyright reasons, we list below the bibliographic information of 12 attached papers, acknowledging the original source of the publications:

(1) Using out-of-language data to improve an under-resourced speech recognizer[25], David Imseng, Petr Motlicek, Hervé Bourlard and Philip N. Garner, in: Speech Communication, 2013 [Ims+13b])

(2) Impact of deep MLP architecture on different acoustic modeling techniques for under-resourced speech recognition[26], David Imseng, Petr Motlicek, Philip N. Garner and Hervé Bourlard, in: Proceedings of the IEEE workshop on Automatic Speech Recognition and Understanding, 2013 [Ims+13a])

(3) Lattice-Free Maximum Mutual Information Training of Multilingual Speech Recognition System[27], Srikanth Madikeri, Banriskhem Khonglah, Sibo Tong, Petr Motlicek, Hervé Bourlard and Daniel Povey, in: Proceedings of Interspeech, pages 4746–4750, ISCA, 2020 [Mad+20a])

(4) Multitask adaptation with Lattice-Free MMI for multi-genre speech recognition of low resource languages[28], Srikanth Madikeri, Petr Motlicek and Hervé Bourlard, in: Proceedings of Interspeech 2021 [MMB21])

(5) Semi-supervised Learning with Semantic Knowledge Extraction for Improved Speech Recognition in Air Traffic Control[29], Ajay Srinivasamurthy, Petr Motlicek, Ivan Himawan, Gyorgy Szaszak, Youssef Oualil and Hartmut Helmke, in: Proceedings of Interspeech 2017, Stockholm, Sweden, pages 2406-2410 [Sri+17])

(6) A COMPARISON OF METHODS FOR OOV-WORD RECOGNITION ON A NEW PUBLIC DATASET[30], Rudolf Braun, Srikanth Madikeri and Petr Motlicek, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, Ontario, Canada, 2021 [BMM21])

(7) English Spoken Term Detection in Multilingual Recordings[31], Petr Motlicek, Fabio Valente and Philip N. Garner, in: Proceedings of Interspeech, ISCA, Makuhari, Japan, 2010 [MVG10])

[25] https://www.sciencedirect.com/science/article/abs/pii/S0167639313000101?via%3Dihub

[26] https://ieeexplore.ieee.org/document/6707752

[27] https://www.isca-speech.org/archive/interspeech_2020/madikeri20_interspeech.html

[28] https://www.isca-speech.org/archive/interspeech_2021/madikeri21_interspeech.html

[29] https://www.isca-speech.org/archive/interspeech_2017/srinivasamurthy17_interspeech.html

[30] https://ieeexplore.ieee.org/document/9415124

[31] https://www.isca-speech.org/archive/interspeech_2010/motlicek10_interspeech.html

(8) A Context-Aware Speech recognition and Understanding System for Air Traffic Control Domain[32], Youssef Oualil, Dietrich Klakow, Gyorgy Szaszak, Ajay Srinivasamurthy, Hartmut Helmke and Petr Motlicek, in: Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop, Okinawa, Japan, 2017 [Oua+17])

(9) Boosting of contextual information in ASR for air-traffic call-sign recognition[33], Martin Kocour, Karel Vesely, Alexander Blatt, Juan Zuluaga-Gomez, Igor Szoke, Jan Cernocky, Dietrich Klakow and Petr Motlicek, in: Proceedings of Interspeech, 2021 [Koc+21b])

(10) Supervised and unsupervised Web-based language model domain adaptation[34], Gwénolé Lecorvé, John Dines, Thomas Hain and Petr Motlicek, in: Proceedings of Interspeech, Portland, Oregon, USA, 2012 [Lec+12])

(11) Conversion of Recurrent Neural Network Language Models to Weighted Finite State Transducers for Automatic Speech Recognition[35], Gwénolé Lecorvé and Petr Motlicek, in: Proceedings of Interspeech, Portland, Oregon, USA, 2012 [LM12])

(12) A two-step approach to leverage contextual data: speech recognition in air-traffic communications[36],[37], Iuliia Nigmatulina, Juan Zuluaga-Gomez, Amrutha Prasad, Seyyed Saeed Sarfjoo and Petr Motlicek, in: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2022 [Nig+21])

[32] https://ieeexplore.ieee.org/document/8268964
[33] https://www.isca-speech.org/archive/interspeech_2021/kocour21_interspeech.html
[34] https://www.isca-speech.org/archive/interspeech_2012/lecorve12_interspeech.html
[35] https://www.isca-speech.org/archive/interspeech_2012/lecorve12b_interspeech.html
[36] https://2022.ieeeicassp.org/papers/accepted_papers.php
[37] http://publications.idiap.ch/index.php/publications/show/4784

5.1 Paper 1: [Ims+13b]

Using out-of-language data to improve an under-resourced speech recognizer

David Imseng[a,b,*], Petr Motlicek[a], Hervé Bourlard[a,b], Philip N. Garner[a]

[a] *Idiap Research Institute, Martigny, Switzerland*
[b] *Ecole Polytechnique Fédérale, Lausanne (EPFL), Switzerland*

Abstract

Under-resourced speech recognizers may benefit from data in languages other than the target language. In this paper, we report how to boost the performance of an Afrikaans automatic speech recognition system by using already available Dutch data. We successfully exploit available multilingual resources through (1) posterior features, estimated by multilayer perceptrons (MLP) and (2) subspace Gaussian mixture models (SGMMs). Both the MLPs and the SGMMs can be trained on out-of-language data. We use three different acoustic modeling techniques, namely Tandem, Kullback–Leibler divergence based HMMs (KL-HMM) as well as SGMMs and show that the proposed multilingual systems yield 12% relative improvement compared to a conventional monolingual HMM/GMM system only trained on Afrikaans. We also show that KL-HMMs are extremely powerful for under-resourced languages: using only six minutes of Afrikaans data (in combination with out-of-language data), KL-HMM yields about 30% relative improvement compared to conventional maximum likelihood linear regression and maximum a posteriori based acoustic model adaptation.

Keywords: Multilingual speech recognition, posterior features, subspace Gaussian mixture models, under-resourced languages, Afrikaans

1. Introduction

Developing a state-of-the-art speech recognizer from scratch for a given language is expensive. The main reason for this is the large amount of data that is usually needed to train current recognizers. Data collection involves large amounts of manual work, not only in time for the speakers to be recorded, but also for annotation of the subsequent recordings. Therefore, the need for training data is one of the main barriers in porting current systems to many languages. On the other hand, large databases already exist for many languages.

Previous studies have shown that automatic speech recognition (ASR) may benefit from data in languages other than the target language only under certain conditions such as there being less than one hour of data for the training language (Imseng et al., 2012a; Qian et al., 2011). Usually, a language with large amounts of training data is used to simulate small amounts of target training data (Imseng et al., 2012a; Qian et al., 2011). For instance (Niesler, 2007) studied the sharing of resources on real under-resourced languages, including Afrikaans, inspired by multilingual acoustic modeling techniques proposed by Schultz and Waibel (2001). However, only marginal ASR performance gains were reported.

Standard ASR systems typically make use of phonemes as subword units to model human speech production. A phoneme is defined as the smallest sound unit of a language that discriminates between a minimal word pair (Bloomfield, 1933, p. 78). Although humans are able to produce a large variety of acoustic sounds, we assume that all those sounds across speakers and languages, share a common acoustic space. We found in previous studies (Imseng et al., 2012a, 2011) that the relation between phonemes of different languages can (1) be learned and (2) be exploited for cross-lingual acoustic model training or adaptation. Posterior features, estimated by multilayer perceptrons (MLPs), are particularly well suited for such tasks. Even though previous posterior feature studies that used more than one hour of target language data reported rather small or no improvements (up to 3.5% relative) (Tòth et al., 2008; Grézl et al., 2011), we successfully used posterior features estimated by MLPs that are trained on similar languages such as English, Dutch and Swiss German to boost the performance of an Afrikaans speech recognizer (Imseng et al., 2012b).

In this paper, we show how to significantly boost the performance of an existing Afrikaans speech recognizer that was trained on three h of within-language data, by using 80 h of Dutch data. We also compare different acoustic modeling techniques and investigate their usefulness if only very limited amounts of within-language data are available.

In our most recent study (Imseng et al., 2012b), we compared two different acoustic modeling techniques for posterior features, namely Tandem (Hermansky et al., 2000) and Kullback-Leibler divergence based hidden Markov models (KL-HMM) (Aradilla et al., 2008). KL-

[*] Corresponding author

HMM and Tandem both exploit multilingual information in the form of posterior features; we found that they benefit from MLPs that were trained on context-dependent targets, but limited ourselves to MLPs with relatively small numbers of context-dependent targets (about 200). In this study however, we further investigate MLPs trained on context-dependent targets and allow ten times more output units. We also investigate a different (and more suitable) cost function for the KL-HMM framework and compare the aforementioned acoustic modeling techniques to subspace Gaussian mixture models (SGMM), conventional maximum likelihood linear regression (MLLR) and maximum a posteriori (MAP) adaptations.

Given three h of Afrikaans data, KL-HMM, Tandem and SGMM successfully exploit 80 h of Dutch data and yield more than 10% relative improvement compared to the conventional HMM/GMM based monolingual recognizer. Furthermore, we also compare the performance of KL-HMM, Tandem, SGMM, MLLR and MAP if only six minutes of Afrikaans data is available. KL-HMM is demonstrated to be particularly well suited to such low amount of data scenarios and outperforms all other acoustic modeling techniques and also MLLR and MAP adaptations.

We first briefly review Tandem, KL-HMM and SGMM in Section 2. In Section 3, we then present the databases that we used for the training of the MLPs and the shared SGMM parameters as described in Section 4, and give an overview over the investigated systems in Section 5. Experiments and results are then given in Section 6 and discussed in Section 7.

2. Acoustic modeling

In this paper, we investigate three different acoustic modeling techniques and also compare them to a conventional HMM/GMM system. The investigated approaches are well suited to exploit out-of-language data. We also compare them to an HMM/GMM system that exploits out-of-language data with the conventional maximum likelihood linear regression (MLLR) approach (Gales, 1998) and with maximum a posteriori (MAP) adaptation (Gauvain and Lee, 1993).

Two of the presented approaches exploit out-of-language data on the feature level, namely Tandem (Hermansky et al., 2000) and Kullback–Leibler divergence based HMM (KL-HMM) (Aradilla et al., 2008). Subspace Gaussian mixture models (SGMM) (Burget et al., 2010) on the other hand exploit out-of-language data on the acoustic model level. The Tandem approach is illustrated in Figure 1, KL-HMM in Figure 2 and SGMM in Figure 3.

The posterior feature based approaches exploit out-of-language information in the form of a Multilayer Perceptron (MLP) which was trained on out-of-language data, whereas the SGMM uses a Universal Background Model (UBM) and shared projection matrices trained on out-of-language data. In the remainder of this section, we will briefly review all three acoustic modeling techniques.

2.1. Feature level

Both posterior feature based approaches involve the training/estimation of two different kind of distributions:

- *Posterior features:* The posterior features are phone class posterior probabilities given the acoustics and estimated with an MLP that can be trained on any auxiliary dataset. Therefore we call it an *auxiliary MLP* and choose an out-of-language dataset with large amounts of available data with which to train. The language of the training data determines the number of output units K (number of phone classes) of the MLP. The phone classes can for example be context-independent monophones or context-dependent triphones. More details about the MLP training are given in Section 4.1.

  Once the MLP is trained, we consider a sequence of T acoustic feature vectors $X = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_T\}$, namely perceptual linear prediction (PLP) features, extracted from within-language data. As seen in Figure 2, the phone class posterior sequence $Z = \{\boldsymbol{z}_1, \dots, \boldsymbol{z}_T\}$ is then estimated with the previously trained auxiliary MLP. To estimate $\boldsymbol{z}_t = (z_t^1, \dots, z_t^K)^\mathsf{T}$, we consider a nine frame temporal context $\{\boldsymbol{x}_{t-4}, \dots, \boldsymbol{x}_t, \dots, \boldsymbol{x}_{t+4}\}$. The described phone class posterior estimation is identical for both posterior feature based acoustic modeling techniques.

- *HMM state distributions:* The HMM states $q^d : d \in \{1, \dots, D\}$ are associated with the target language. Each phone (mono- or tri-phone) of the target language is modeled with three states, thus the total number of states D is equal to three times the number of phones of the target language.

  The HMM state distributions consist of emission and transition probabilities. Based on anecdotal knowledge, we fix the transition probabilities a_{ij} for both posterior feature based acoustic modeling techniques (see Figures 1 and 2). The emission probabilities however are modeled differently for Tandem and KL-HMM. As we will describe below, Tandem (Section 2.1.1) uses Gaussian mixtures and KL-HMM (Section 2.1.2) uses a categorical distribution. The emission probabilities are trained from within-language data only. Here, we assume that we have access to a limited amount of within-language data.

2.1.1. Tandem

The conventional Tandem approach models the emission probabilities of the HMM states q^d with mixtures of Gaussians. Figure 1 illustrates the HMM associated with a three-state-phone (q^i, q^j, q^k). To model the emission probabilities with Gaussians, the posterior features z_t

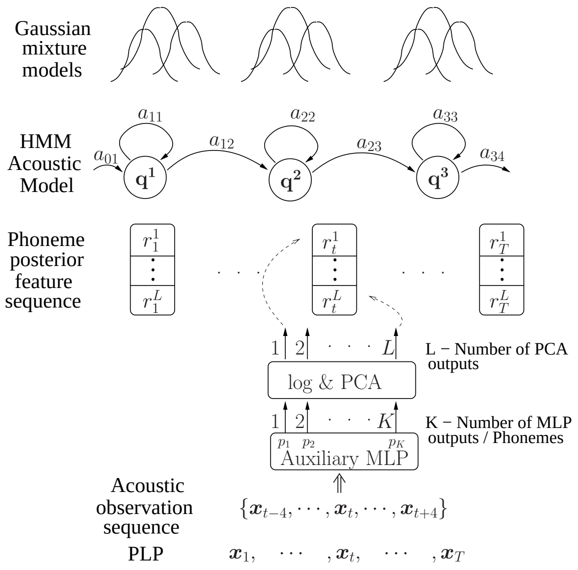

Gaussian mixture models
HMM Acoustic Model
a_{11} a_{22} a_{33}
a_{01} a_{12} a_{23} a_{34}
$\mathbf{q}^1$ $\mathbf{q}^2$ $\mathbf{q}^3$
Phoneme posterior feature sequence
r_1^1 r_1^L r_t^1 r_t^L r_T^1 r_T^L
1 2 · · · L
L − Number of PCA outputs
log & PCA
1 2 · · · K
K − Number of MLP outputs / Phonemes
p_1 p_2 p_K
Auxiliary MLP
Acoustic observation sequence
$\{\boldsymbol{x}_{t-4}, \cdots, \boldsymbol{x}_t, \cdots, \boldsymbol{x}_{t+4}\}$
PLP
$\boldsymbol{x}_1, \cdots, \boldsymbol{x}_t, \cdots, \boldsymbol{x}_T$

Figure 1: Tandem - the emission probabilities of the HMM states are modeled with Gaussian mixtures and the MLP output is post-processed. For more details, see Section 5.4.

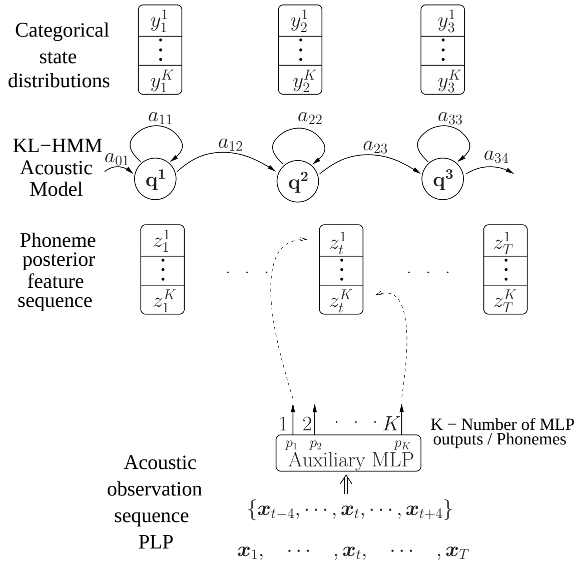

Categorical state distributions
y_1^1 y_1^K y_2^1 y_2^K y_3^1 y_3^K
KL−HMM Acoustic Model
a_{11} a_{22} a_{33}
a_{01} a_{12} a_{23} a_{34}
$\mathbf{q}^1$ $\mathbf{q}^2$ $\mathbf{q}^3$
Phoneme posterior feature sequence
z_1^1 z_1^K z_t^1 z_t^K z_T^1 z_T^K
1 2 · · · K
K − Number of MLP outputs / Phonemes
p_1 p_2 p_K
Auxiliary MLP
Acoustic observation sequence
$\{\boldsymbol{x}_{t-4}, \cdots, \boldsymbol{x}_t, \cdots, \boldsymbol{x}_{t+4}\}$
PLP
$\boldsymbol{x}_1, \cdots, \boldsymbol{x}_t, \cdots, \boldsymbol{x}_T$

Figure 2: KL-HMM - the emission probabilities are modeled with categorical distributions and the MLP output can directly be used. More details can be found in Section 5.5.

need to be post-processed. More specifically, the log-phone class posteriors are decorrelated with a principal component analysis (PCA). The transformation matrix can be estimated on within-language data. Usually, the resulting feature vector $\boldsymbol{r}_t = (r_t^1, \ldots, r_t^L)^\intercal$, has a reduced dimensionality L.

2.1.2. Kullback–Leibler divergence based HMM

As illustrated in Figure 2, a KL-HMM is a particular form of HMM in which the emission probability of state q^d is parametrized by a categorical distribution $\boldsymbol{y}_d = (y_d^1, \ldots, y_d^K)^\intercal$, where K is the dimensionality of the features. A categorical distribution is a multinomial distribution from which only one sample is drawn. In contrast to Tandem that uses Gaussian mixtures and therefore needs the post-processed features $\boldsymbol{r}_t$, the categorical distributions can directly be trained from phone class posterior probabilities $\boldsymbol{z}_t$.

Kullback and Leibler introduced the term *discrimination information* (Kullback and Leibler, 1951; Kullback, 1987) which is nowadays often referred to as the *Kullback–Leibler distance*[1], defined by Cover and Thomas (1991). The divergence of Kullback and Leibler (1951) is today referred to as the symmetric variant of the KL divergence. Aradilla et al. (2008) proposed multiple KL divergence based local scores for KL-HMM training and decoding. In recent studies, we used the symmetric variant of the KL divergence. However, recently we found that the asymmetric KL divergence $KL(x||y)$ is in fact more robust. This is also intuitively reasonable in that the underlying acoustic modeling problem is not symmetric since we observe the posterior features and train the categorical distributions. Therefore, we use the following Kullback–Leibler based distance as local score in this study:

$$d(\boldsymbol{z}_t, \boldsymbol{y}_d) = \sum_{k=1}^{K} z_t^k \log \frac{z_t^k}{y_d^k}. \tag{1}$$

A detailed description of training and decoding algorithms based on the symmetric variant of the KL divergence can be found in (Imseng et al., 2012a). In this paper we use the asymmetric KL divergence as given in (1). For clarity, we briefly review the training and decoding algorithms.

Training

The categorical distributions $Y = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_D\}$ can be learned using an iterative Viterbi segmentation-optimization scheme. The cost function can be defined by integrating the local score, given in (1), over time t and states q^d, resulting in

$$\mathcal{F}(Z, Y) = \sum_{t=1}^{T} \sum_{d=1}^{D} d(\boldsymbol{z}_t, \boldsymbol{y}_d)\delta_t^d, \tag{2}$$

where the Kronecker delta δ_t^d is defined as:

$$\delta_t^d = \begin{cases} 1, & \text{if } \boldsymbol{x}_t \text{ is associated with state } q^d \\ 0, & \text{otherwise.} \end{cases}$$

To associate each $\boldsymbol{x}_t$ with one of the states, the HMM aligns the phone class posterior probabilities Z with the states by minimizing $\mathcal{F}(Z, Y)$, given in (2).

Each $\boldsymbol{z}_t$ is then used to update a particular categorical distribution $\boldsymbol{y}_d$. To minimize $\mathcal{F}(Z, Y)$ subject to $\sum_{k=1}^{K} y_d^k = 1$, we take the partial derivative with respect to each variable y_d^k and set it to zero to find the minimum.

[1] In reality, usually it is referred to as a divergence rather than a distance because it is not a metric.

Then, we introduce the Lagrange multipliers λ to enforce the sum to one constraint:

$$\frac{\partial}{\partial y_d^k}\left[\mathcal{F}(Z,Y)+\lambda\left(\sum_{k=1}^{K} y_d^k - 1\right)\right] = 0. \tag{3}$$

Solving (3) yields:

$$y_d^k = \frac{1}{\lambda}\sum_{\forall t^*} z_t^k, \tag{4}$$

where the sum extends over all t^* such that $\boldsymbol{x}_{t^*}$ is associated with state q_d. Solving (4) for λ yields:

$$\lambda = \sum_{\forall t^*}\sum_{k=1}^{K} y_d^k = \sum_{\forall t^*} 1 = T_d,$$

where T_d stands for the number of frames associated with state q_d. We thus obtain:

$$y_d^k = \frac{1}{T_d}\sum_{\forall t^*} z_t^k. \tag{5}$$

Decoding

During decoding, we minimize:

$$\mathcal{F}_{\mathcal{Q}}(Z,Y) = \min_{\mathcal{Q}}\sum_{t=1}^{T}\left[d(\boldsymbol{z}_t,\boldsymbol{y}_{q_t}) - \log a_{q_{t-1}q_t}\right], \tag{6}$$

where $\mathcal{Q} = \{q_1,\ldots,q_T\}$ stands for all allowed state paths and $\boldsymbol{y}_{q_t}$ is the categorical distribution associated with q_t, the state at time t. The transition probabilities $a_{q_{t-1}q_t}$ are fixed.

2.2. Acoustic model level

In addition to feature level, out-of-language data can also be directly exploited on the acoustic model level to improve ASR performance. In this study we employ SGMMs as an acoustic modeling technique exploiting out-of-language data. Similar to feature level, HMM state distributions associated with the target language are estimated. The transition probabilities are fixed and the emmission probabilities are modeled using probability density function in an SGMM manner.

2.2.1. Subspace Gaussian mixture model (SGMM)

In the SGMM acoustic modeling approach, each speech state associated with an HMM is modeled by a GMM, as is the case for conventional HMM/GMMs. However, the GMMs are not the parameters of the model. Instead, each HMM state q^d (where d represents a state index) is associated with a vector $\boldsymbol{v}_d = (v_d^1,\ldots,v_d^S)^\mathsf{T}$, where S is usually similar to the dimension of the acoustic speech features. Mathematically, the SGMM model can be described as follows (Povey et al., 2010):

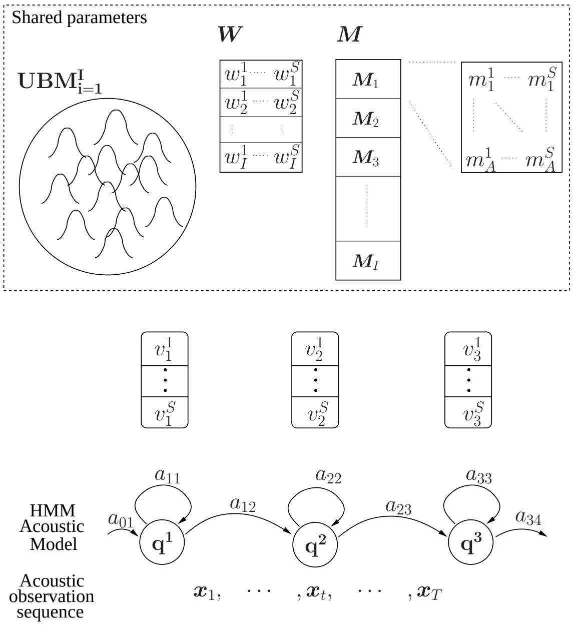


Figure 3: SGMM - the emission probabilities of each context-dependent HMM-state q_d are modeled by GMM. Each HMM-state is parametrized by a vector $\boldsymbol{v}_d$. The parameters $\mathbf{M}$ and $\mathbf{W}$ are globally shared.

$$\begin{aligned}
p(\boldsymbol{x}_t|q^d) &= \sum_{i=1}^{I}\omega_i^d\mathcal{N}(x_t;\boldsymbol{\mu}_{i,d},\boldsymbol{\Sigma}_i), &(7)\\
\boldsymbol{\mu}_{i,d} &= \mathbf{M}_i\boldsymbol{v}_d, &(8)\\
\omega_i^d &= \frac{\exp(\boldsymbol{w}_i\cdot\boldsymbol{v}_d)}{\sum_{l=1}^{I}\exp(\boldsymbol{w}_l\cdot\boldsymbol{v}_d)}, &(9)
\end{aligned}$$

where $\boldsymbol{x}_t \in R^A$ denotes feature vector, q^d is the HMM-state, and $\boldsymbol{v}_d \in R^S$ is the state-specific vector. The model in each HMM state is represented by a simple GMM with I Gaussians, mixture weights $\boldsymbol{\omega}^d = (\omega_1^d,\ldots,\omega_I^d)^\mathsf{T}$, means $\boldsymbol{\mu}_{i,d}$, and covariances $\boldsymbol{\Sigma}_i$. $\boldsymbol{\Sigma}_i$ are shared across the states. The state-specific vectors $\boldsymbol{v}_d \in R^S$ together with the globally shared parameters $\boldsymbol{M} = (\boldsymbol{M}_1,\ldots,\boldsymbol{M}_I)^\mathsf{T}$ and $\boldsymbol{W} = (\boldsymbol{w}_1,\ldots,\boldsymbol{w}_I)^\mathsf{T}$ with $\boldsymbol{w}_i = (w_i^1,\ldots,w_i^S)$ are used to derive the means and mixture weights representing the given HMM state. For the initialization of the SGMM, a generic GMM with I Gaussians, denoted as UBM, modeling all the speech is used. SGMM acoustic modeling is depicted in Figure 3 for a single HMM.

Note that the equations above assume (without loss of generality) one state-specific vector $\boldsymbol{v}_d$ to be assigned to each HMM-state. However, as done for the experiments in this study, we can model each state with a mixture of sub-states (Povey et al., 2011), each having its own sub-state specific vector $\boldsymbol{v}_{d_j}$, where $j = 1,\ldots,J_d$ with J_d being the number of sub-states of state d. In that case, we also need to estimate the mixture weights c_j for each sub-state.

ID	Language	Number of phonemes	Amount of training data
AF	Afrikaans	38	3 h
CGN	Dutch	47	81 h

Table 1: Summary of the different languages with number of phonemes and amount of available training data.

3. Databases

We used data from Afrikaans and Dutch as summarized in Table 1. In this section, we describe the two databases.

3.1. LWAZI

The Afrikaans data is available from the LWAZI corpus provided by the Meraka Institute, CSIR, South Africa[2] and described by Barnard et al. (2009). The database consists of 200 speakers, recorded over a telephone channel at 8 kHz. Each speaker produced approximately 30 utterances, where 16 were randomly selected from a phonetically balanced corpus and the remainder consisted of short words and phrases.

The Afrikaans database comes with a dictionary (Davel and Martirosian, 2009) that defines the phoneme set containing 38 phonemes (including silence). The dictionary that we used contained 1585 different words. The HLT group at Meraka provided us with the training and test sets. In total, about three hours of training data and 50 minutes of test data is available (after voice activity detection).

Since we did not have access to an appropriate language model, we trained a bi-gram phoneme model on the training set and only report phoneme accuracies in this study. The bi-gram phoneme model learned the phonotactic constraints of the Afrikaans language and has a phoneme perplexity of 14.5 on the training set and 14.7 on the test set.

3.2. Corpus Gesproken Nederlands

Heeringa and de Wet (2008) reported that standard Dutch seems to be the best language from which to borrow acoustic data for the development of an Afrikaans ASR system. Our studies confirmed that hypothesis (Imseng et al., 2012b). Therefore, we used data of the Spoken Dutch Corpus (Corpus Gesproken Nederlands, CGN) (Oostdijk, 2000) that contains standard Dutch pronounced by more than 4000 speakers from the Netherlands and Flanders. The database is divided into several subsets and we only used “Corpus o” because it contains phonetically aligned *read* speech data pronounced by 324 speakers from the Netherlands and 150 speakers from Flanders. “Corpus o” uses 47 phonemes and contains 81 h of data after the deletion of silence segments that are longer than one second. It was recorded at 16 kHz, but since we use the data to perform ASR on Afrikaans, we downsampled it to 8 kHz prior to feature extraction.

ID	Language	Number of output units	Frame accuray on validation set
AF	Afrikaans	187	43.8%
CGN	Dutch	1789	56.5%

Table 2: Summary of the MLPs with number of output units and frame accuracy on the cross-validation set.

4. Multilingual boosting strategies

In this section, we describe how out-of-language data is exploited in case of feature-level and acoustic model-level adaptation.

4.1. Feature level approach

For each language (Afrikaans and Dutch), we trained an MLP from 39 Mel-Frequency Perceptual Linear Prediction (MF-PLP) features ($C0-C12+\Delta+\Delta\Delta$) in a nine frame temporal-context (four preceding and following frames), extracted with the HTS variant (Zen et al., 2007) of the HTK toolkit. The number of parameters in each MLP was set to 10% of the number of available training frames, to avoid overfitting. We used Quicknet (Johnson, 2005) software to train the MLPs.

We have already shown that systems that use MLPs which are trained on context-dependent targets (triphones) outperform MLPs trained on context-independent monophones (Imseng et al., 2012b). Therefore, we trained both MLPs on triphone targets. To obtain triphone targets, we developed a standard HMM/GMM system with all the training data for both languages and used a standard decision tree approach to tie rare triphones. More specifically, we used the minimum description length criterion to automatically determine the number of tied triphones for each language independently (Shinoda and Watanabe, 1997). As described by Shinoda and Watanabe (1997), the MDL criterion has a hyper-parameter, c, which controls the weight of the term that penalizes models with large amounts of triphones. We tuned c on the Afrikaans database and fixed it to 16 (for both databases). The HMM/GMM systems were then used to align the training data in terms of tied triphones. We used 90% of the training set for training and 10% for cross-validation to stop training. Table 2 shows the number of output units (tied triphones) per MLP and frame accuracy on the cross-validation set.

4.2. Acoustic model level approach

To exploit out-of-language data, the SGMM model parameters can be divided into HMM-state specific and shared parameters, as visualized in Figure 3. As proposed by Burget et al. (2010), $\boldsymbol{M}$ and $\boldsymbol{W}$ projection matrices together with UBM can be perceived as shared (language-independent) and can therefore be trained using large amounts of data from different languages. As already mentioned in Section 2.2.1, we use several sub-states per HMM-state. The sub-state-specific vectors $\boldsymbol{v}_{\boldsymbol{d}_j}$ as well

[2] `http://www.meraka.org.za/hlt`

Afrikaans	Dutch
ɑː	ɑ
ae	ɛ
oe	ʏ
øː	ø
ɦ	h

Table 3: The Afrikaans phonemes without a matching Dutch seed model (same IPA symbol not present in the Dutch phoneme set) are given in the left column. The corresponding manually chosen Dutch seed models are listed in the right column.

as the mixture weights c_j are trained on within-language data.

5. Systems

In this section, we will describe the systems that we investigated to study the exploitation of out-of-language data in the framework of under-resourced ASR. We will compare the performance of the Tandem approach with the performance of KL-HMM and SGMM. Furthermore, we will also compare the proposed systems to an HMM/GMM baseline only trained on within-language data and to an HMM/GMM system trained on Dutch and then adapted to Afrikaans by using MLLR and MAP.

5.1. HMM/GMM

Each context-dependent triphone is modeled with three states (q^i, q^j, q^k). As usually done, we first train context-independent monophone models that serve as seed models for the context-dependent triphone models. We use eight Gaussians per state to model the emission probabilities. To balance the number of parameters with the amount of available training data, we apply conventional state tying with a decision tree that is based on the minimum description length principle (Shinoda and Watanabe, 1997). Training and decoding is performed with HTS.

5.2. Maximum likelihood linear regression (MLLR)

To evaluate whether an under-resourced language could be accommodated by linear transforms, we first train a triphone HMM/GMM system on the Dutch data. Each triphone state is modeled with 16 Gaussians. We investigate the standard MLLR and use a regression tree that allows up to 32 regression classes.

For most Afrikaans phonemes, we use the corresponding Dutch phoneme, represented with the same IPA symbol, as a seed model for MLLR. However, not all the Afrikaans phonemes are also present in the Dutch phoneme set. The Afrikaans phonemes without matching Dutch seed model are given in Table 3 together with the respective manually chosen Dutch seed model. Furthermore, since the diphthongs iə, uə, əu, əi are not present in the Dutch phoneme set, we split them into individual phonemes (monophthongs).

5.3. Maximum a posteriori (MAP) adaptation

Since Köhler (1998) has shown that MAP adaptation is suitable for cross-lingual acoustic model adaptation, we also evaluate MAP adaptation. More specifically, the mean $\mu_{j,m}$ of mixture component m and state j is adapted as follows:

$$\hat{\boldsymbol{\mu}}_{j,m} = \frac{N_{j,m}}{N_{j,m}+\tau}\boldsymbol{\mu}^A_{j,m} + \frac{\tau}{N_{j,m}+\tau}\boldsymbol{\mu}^D_{j,m}, \qquad (10)$$

where $N_{j,m}$ is the occupation likelihood of the Afrikaans data, τ a parameter to tune, $\boldsymbol{\mu}^A$ the mean of the Afrikaans data and $\boldsymbol{\mu}^D$ the mean of the Dutch data.

As seed models, we used the same Dutch context-dependent HMM/GMM models as in Section 5.2. For Afrikaans phonemes without a matching Dutch seed model, we again mapped phonemes as explained in Section 5.2 and Table 3.

5.4. Tandem

Similar to the conventional HMM/GMM system, for the Tandem system, we train context-independent monophone models that serve as seed models for the three-state context-dependent triphone models. We use eight Gaussians per state to model the emission probabilities and use PCA for decorrelation. PCA can also be used to reduce the dimensionality to, for example, 30, as is typically done (Qian et al., 2011; Grézl et al., 2011). In recent studies, we have shown that the dimensionality of the feature vectors greatly affects the performance of the Tandem system (Imseng et al., 2012b). Furthermore, we observed that preserving 99% of the variance yielded similar results to using all the dimensions (Imseng et al., 2012b). Therefore, in this study, we fix the dimensionality such that 99% of the variance is preserved (note that the dimensionality of different systems varies and is given in Tables 4, 5 and 6).

To balance the number of parameters with the amount of available training data, we also use a decision tree that is based on the minimum description length principle (Shinoda and Watanabe, 1997).

5.5. KL-HMM

As for HMM/GMM and Tandem, for the KL-HMM system, we train context-independent monophone models that serve as seed models for the three-state context-dependent triphone models.

For KL-HMM, we applied a decision tree clustering reformulated as dictated by the KL criterion (Imseng et al., 2012c). Since it is not obvious how to apply the minimum description length principle to the modified clustering approach, we tuned the threshold that determines the number of tied states on the development set.

System	Feature dimension	Number of tied states	Phoneme accuracy
HMM/GMM	39	1447	61.2 %
KL-HMM	187	15207	60.6 %
Tandem	48	1846	*64.7* %
SGMM	39	2000	**65.5** %

Table 4: Using 3 h of Afrikaans data to build a monolingual ASR system. Acoustic modeling techniques are described in Section 5. The best result is marked bold; italic numbers point to results that are not significantly worse.

5.6. SGMM

The training of SGMMs is also done from context-independent monophone models that serve as seed models for the three-state context-dependent triphone models.

Decision tree clustering was done automatically, after having specified the number of leaves to be similar to the Tandem system. The UBM has $I = 500$ Gaussians and the dimensionality of the substate phone-specific vectors, S, was fixed to 50.

6. Evaluation

In this section, we analyze the performance of the different systems. We always apply the same bi-gram phoneme model as described in Section 3.1 and report Afrikaans phoneme accuracies on the test set (about 50 min of data). The bi-gram phoneme model scaling factor was determined for each system independently on the cross-validation set (see Section 4.1). In general, we expect that the exploitation of Dutch data will improve the Afrikaans ASR performance. For all the significance tests, we used the bootstrap estimation method (Bisani and Ney, 2004) and a confidence interval of 95%.

6.1. Afrikaans data only

For the first set of experiments, we only used the Afrikaans training set (3 h of data) for the training of the global and local parameters. More specifically, the MLP for the posterior feature extraction as well as the globally shared SGMM parameters were trained on three hours of Afrikaans (see Table 2 for MLP details). In previous studies (Povey et al., 2010), SGMM outperformed HMM/GMM when 15 h of training data was used. We hypothesize that SGMM also outperforms conventional HMM/GMM if only three hours of data is available for training. Furthermore, Tandem outperformed conventional HMM/GMM and KL-HMM systems if three hours of Afrikaans data was available for training (Imseng et al., 2012b).

Table 4 shows the results. Note that the baseline results reported by van Heerden et al. (2009), 63.1% phoneme accuracy, were the first set of results obtained for the data and the official train and test set were compiled after the official database release. Personal communication with the HLT group at Meraka confirmed that the lower performance of our baseline can be attributed to the different data partitioning[3].

The results in Table 4 confirm our hypotheses. On Afrikaans data only, SGMM performs best, followed by Tandem. Bold numbers in tables mark the best result (column-wise) and italic numbers are not significantly different from the best performance. KL-HMM and the HMM/GMM baseline perform significantly worse than SGMM and Tandem.

Table 4 also shows the feature dimensionality of the employed acoustic modeling techniques. HMM/GMM and SGMM are both based on acoustic features (39 dimensions). KL-HMM uses the raw output of the Afrikaans MLP. For the Tandem system however, recall that the posterior features need to be post-processed. Keeping 99% of the variance after PCA results in a 48-dimensional feature vector.

The number of tied states, also shown in Table 4, for HMM/GMM and for Tandem were automatically determined with the MDL criterion. Based on anecdotal knowledge, we fixed the number of tied states for the SGMM system similar to the number of tied states for the Tandem system. The number of tied states for the KL-HMM was tuned on the cross-validation set. Since the categorical distributions of the KL-HMM can be trained with very few data, modeling each triphone state separately performed best. Hence, the decision tree was only used to synthesize unseen triphone contexts during testing.

Due to the extremely high number of states of the KL-HMM system compared to the other systems, the KL-HMM system has the most parameters of the four systems given in Table 4. To increase the number of parameters of the other systems, we increased the number of Gaussians per state for the HMM/GMM as well as for the Tandem system to 16 and doubled the number of sub-states of the SGMM system. However, none of the performances improved.

6.2. Auxiliary Dutch data

For the second set of experiments, we used the Dutch data to train the MLP as well as the globally-shared SGMM parameters. We also trained Dutch seed models for the MLLR and MAP adaptation. The Afrikaans data was used to train the HMM distributions (KL-HMM and Tandem), the sub-state phone-specific vectors $\mathbf{v}_d$ and sub-state mixture weights $\mathbf{c}_j$ (SGMM) and the MLLR adaptation. MAP adaptation was applied as described in (10) and τ was tuned on the development set (see Table 5).

Since three hours seems to be a reasonable amount of training data, we also simulated very low-resourced languages and evaluated three different scenarios: six minutes of data, one hour of data and three hours of data.

[3] The HLT group now also uses the partitioning that we used in this paper and report a lower performance.

System	Feat.	6 min			1 h			3 h		
	dim.	TS	τ	PA [%]	TS	τ	PA [%]	TS	τ	PA [%]
HMM/GMM	39	116	—	38.6	594	—	55.3	1447	—	61.2
MLLR	39	—	—	41.3	—	—	44.4	—	—	44.7
MAP	39	11357	15	39.4	11357	5	46.9	11357	1	50.6
KL-HMM	1789	635	—	**53.1**	13308	—	**61.5**	15207	—	*67.3*
Tandem	286	114	—	41.0	537	—	*61.3*	1846	—	*68.2*
SGMM	39	150	—	40.2	600	—	*60.4*	2000	—	**68.5**

Table 5: Exploiting Dutch data to improve an Afrikaans ASR system. The different acoustic modeling techniques are described in Section 5. TS stands for the number of tied states, PA for phoneme accuracy and τ is the parameter of the MAP adaptation. Best results of each PA column are marked bold; italic numbers point to results that are not significantly worse.

For the sake of comparison, we also evaluated a conventional HMM/GMM system for each scenario. We hypothesize, that KL-HMM performs best for very low amounts of data because we have seen this behavior in previous similar evaluations of KL-HMM (Imseng et al., 2012c). If three hours of data is available, we expect that KL-HMM, Tandem and SGMM are successfully exploiting the out-of-language data and performing similarly well.

Table 5 confirms our hypotheses. The HMM/GMM (only trained on Afrikaans) is clearly outperformed by KL-HMM, Tandem and SGMM, hence all three systems successfully exploit out-of-language information. MLLR and MAP, however, only perform better than HMM/GMM if six minutes of Afrikaans data are available. Note that both approaches are bound to phoneme sets. Köhler (1998) for example had for each phoneme a multilingual seed model that was trained from data associated with a matching IPA symbol. In our case however, we needed to manually map several Afrikaans phoneme models as discussed in Table 3. If there is 1 h or more data available, MAP outperforms MLLR.

For the three hours as well as the one hour scenario, SGMM, KL-HMM and Tandem all perform statistically the same. While SGMM is the most suitable acoustic modeling technique if we train only on within-language data, KL-HMM (which was performing significantly worse in Table 4) benefits most from out-of-language data and seems to be particularly well suited to exploit out-of-language information on this database. Furthermore, KL-HMM using six minutes of data performs almost as well as a conventional monolingual HMM/GMM system using one hour of data. In the case of the SGMM, results are slightly worse than expected. We suppose that the dimensionality of the sub-state-specific vectors is probably too high for only six minutes of data.

6.3. Within- and out-of-language data

We have already shown that properly combining acoustic information from multiple similar languages can boost the performance. Therefore, we hypothesize that the performance can be improved if we concatenate the output of both MLPs or train the shared SGMM parameters on both languages. The concatenated MLP outputs were renormalized to guarantee that the feature vectors can be interpreted as posterior distributions, as assumed by the KL-HMM. For the Tandem systems, we post-process the normalized vectors as already described in Section 5.4. For SGMM, we trained the shared parameters with the data of both languages.

System	Feature dimension	Phoneme accuracy
KL-HMM	1976	**68.8** %
Tandem	308	*68.4* %
SGMM	39	*68.6* %

Table 6: Using the Dutch and Afrikaans MLP (KL-HMM and Tandem) and use Dutch and Afrikaans data to train the shared parameters (SGMM). The best result is marked bold; italic numbers point to results that are not significantly worse.

However, Table 6 shows that the results only marginally improve for Tandem and SGMM. For KL-HMM, they improve by 1.5% absolute. KL-HMM performs best but not statistically differently from the other systems.

7. Discussion

The results in Section 6 have shown that (a) out-of-language data improved an existing Afrikaans speech recognizer and (b) KL-HMM outperforms all other approaches if only 6 min of Afrikaans data are available. In this section, we discuss the two conclusions.

7.1. Out-of-language data

The systems in Table 6 perform significantly better than the HMM/GMM baseline that does not use Dutch data (see Table 4). We hypothesize that Dutch data mostly improve recognition accuracy of phonemes for which the Afrikaans dataset does not contain much training data. Figure 4 shows the relative phoneme accuracy change per phoneme of the systems given in Table 6 with respect to the HMM/GMM baseline that does not use Dutch data. The phonemes on the x-axis are sorted according to their frequency in the Afrikaans training data with the most frequent phonemes on the left. Figure 4 appears to confirm our hypothesis.

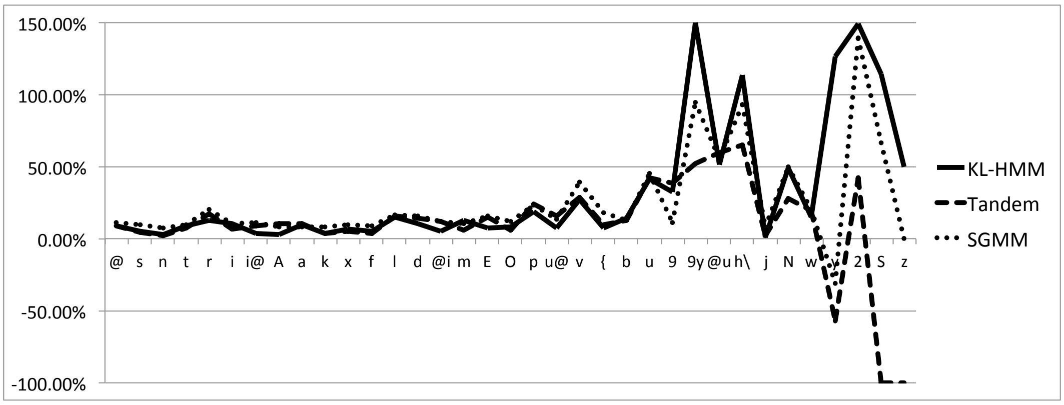


Figure 4: Relative phoneme accuracy change per phoneme of the systems shown in Table 6 with respect to the monolingual HMM/GMM baseline system. The phonemes on the x-axis are sorted according to their frequency in the Afrikaans training data (most frequent phoneme on the left).

7.2. KL-HMM

Even though we performed an extensive error analysis, there was no clear evidence for which phonemes KL-HMM yields most improvement compared to the other modeling techniques. Rather, KL-HMM consistently improves the recognition accuracy across all phonemes. We attribute the improvement to the sophisticated acoustic modeling and the constrained optimization space that are particularly well suited for low amount of data scenarios.

8. Conclusion and future work

We successfully exploited Dutch data and boosted a monolingual speech recognizer that was trained on three h of Afrikaans data. We compared KL-HMM, Tandem, SGMM, MLLR as well as MAP and found that KL-HMM, Tandem and SGMM successfully exploit out-of-language data if at least one hour of within-language data are available. If only six minutes of data are available, KL-HMM outperforms all other acoustic modeling techniques including MLLR and MAP.

Furthermore, we found that if three h of within-language data and 80 h of out-of-language data are available, the proposed systems yield 12% relative improvement compared to a conventional HMM/GMM system only using within-language data. If only six minutes of within-language data and 80 h of out-of-language data are available, KL-HMM performs relatively about 30% better than MLLR and MAP .

We exploited multilingual information on the feature level by applying simple concatenation of MLP outputs. In future, we plan to explore different methods to combine the output of several MLPs. Furthermore, we also exploited multilingual information on the acoustic modeling level. To investigate whether the two approaches are complementary, we plan to implement an SGMM system based on posterior features.

9. Acknowledgement

This research was supported by the Swiss NSF through the project Interactive Cognitive Systems (ICS) under contract number 200021_132619/1 and the National Centre of Competence in Research (NCCR) in Interactive Multimodal Information Management (IM2) `http://wwww.im2.ch`.

The authors are grateful to the HLT group at Meraka, and especially Dr. Febe de Wet, for providing us with the training and test sets as well as the Afrikaans dictionary.

References

D. Imseng, H. Bourlard, P. N. Garner, Using KL-divergence and multilingual information to improve ASR for under-resourced languages, in: Proc. of ICASSP, 4869–4872, 2012a.

Y. Qian, J. Xu, D. Povey, J. Liu, Strategies for Using MLP based Features with Limited Target-Language Training Data, in: Proc. of ASRU, 354–358, 2011.

T. Niesler, Language-dependent state clustering for multilingual acoustic modelling, Speech Communication 49 (2007) 453–463.

T. Schultz, A. Waibel, Language-independent and language-adaptive acoustic modeling for speech recognition, Speech Communication 35 (2001) 31–51.

L. Bloomfield, Language, New York: Holt, 1933.

D. Imseng, H. Bourlard, J. Dines, P. N. Garner, M. Magimai.-Doss, Improving non-native ASR through stochastic multilingual phoneme space transformations, in: Proc. of Interspeech, 537–540, 2011.

L. Tòth, J. Frankel, G. Gosztolya, S. King, Cross-lingual portability of MLP-based tandem features - a case study for English and Hungarian., in: Proc. of Interspeech, 2695–2698, 2008.

F. Grézl, M. Karafiát, M. Janda, Study of Probabilistic and Bottle-Neck Features in Multilingual Environment, in: Proc. of ASRU, 359–364, 2011.

D. Imseng, H. Bourlard, P. N. Garner, Boosting under-resourced speech recognizers by exploiting out of language data - Case study on Afrikaans, in: Proceedings of the 3rd International Workshop on Spoken Languages Technologies for Under-resourced Languages, 60–67, 2012b.

H. Hermansky, D. Ellis, S. Sharma, Tandem connectionist feature extraction for conventional HMM systems, in: Proc. of ICASSP, III–1635–1638, 2000.

G. Aradilla, H. Bourlard, M. Magimai-Doss, Using KL-Based Acoustic Models in a Large Vocabulary Recognition Task, in: Proc. of Interspeech, 928–931, 2008.

M. Gales, Maximum likelihood linear transformations for HMM-based speech recognition, Computer Speech and Language 12 (2) (1998) 75 – 98.

J.-L. Gauvain, C.-H. Lee, Speaker adaptation based on MAP estimation of HMM parameters, in: Proc. of ICASSP, vol. 2, 558–561, 1993.

L. Burget, et al., Multilingual Acoustic Model for Speech Recognition based on Subspace Gaussian Mixture Models, in: Proc. of ICASSP, 4334–4337, 2010.

S. Kullback, R. A. Leibler, On Information and Sufficiency, The Annals of Mathematical Statistics 22 (1) (1951) 79–86, http://projecteuclid.org/DPubS/Repository/1.0/Disseminate?view=body&id=pdf_1&handle=euclid.aoms/1177729694.

S. Kullback, The Kullback-Leibler Distance, The American Statistician 41 (4) (1987) 340–341, in Letters to the Editor.

T. Cover, J. Thomas, Elements of information theory, Wiley, New York, 1991.

D. Povey, et al., Subspace Gaussian Mixture Models for Speech Recognition, in: Proc. of ICASSP, 4330–4333, 2010.

D. Povey, M. Karafiát, A. Ghoshal, P. Schwarz, A Symmetrization of the Subspace Gaussian Mixture Model, in: Proc. of ICASSP, 4504–4507, 2011.

E. Barnard, M. Davel, C. van Heerden, ASR Corpus design for resource-scarce languages, in: Proc. of Interspeech, 2847–2850, 2009.

M. Davel, O. Martirosian, Pronunciation dictionary development in resource-scarce environments, in: Proc. of Interspeech, 2851–2854, 2009.

W. Heeringa, F. de Wet, The origin of Afrikaans pronunciation: a comparison to west Germanic languages and Dutch dialects, in: Proc. of the Conf. of the Pattern Recognition Association of South Africa, 159–164, www.let.rug.nl/heeringa/dialectology/papers/prasa08.pdf, 2008.

N. Oostdijk, The Spoken Dutch Corpus. Overview and first evaluation., in: In Proceedings of the Second International Conference on Language Resources and Evaluation, vol. II, 887–894, 2000.

H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. Black, K. Tokuda, The HMM-based speech synthesis system version 2.0, http://hts.sp.nitech.ac.jp/, 2007.

D. Johnson, Quicknet, http://www.icsi.berkeley.edu/Speech/qn.html, 2005.

K. Shinoda, T. Watanabe, Acoustic modeling based on the MDL principle for speech recognition, in: Proc. of Eurospeech, vol. I, 99–102, 1997.

J. Köhler, Language adaptation of multilingual phone models for vocabulary independent speech recognition tasks, in: Proc. of ICASSP, vol. 1, 417–420, 1998.

D. Imseng, J. Dines, P. Motlicek, P. N. Garner, H. Bourlard, Comparing different acoustic modeling techniques for multilingual boosting, in: Proc. of Interspeech, 2012c.

M. Bisani, H. Ney, Bootstrap estimates for confidence intervals in ASR performance evaluation, in: Proc. of ICASSP, vol. 1, I–409–412, 2004.

C. van Heerden, E. Barnard, M. Davel, Basic speech recognition for spoken dialogues, in: Proc. of Interspeech, 3003–3006, 2009.

5.2 Paper 2: [Ims+13a]

IMPACT OF DEEP MLP ARCHITECTURE ON DIFFERENT ACOUSTIC MODELING TECHNIQUES FOR UNDER-RESOURCED SPEECH RECOGNITION

David Imseng[1]*, Petr Motlicek*[1]*, Philip N. Garner*[1]*, Hervé Bourlard*[1,2]

[1]Idiap Research Institute, Martigny, Switzerland
[2]Ecole Polytechnique Fédérale, Lausanne (EPFL), Switzerland
{dimseng,motlicek,pgarner,bourlard}@idiap.ch

ABSTRACT

Posterior based acoustic modeling techniques such as Kullback–Leibler divergence based HMM (KL-HMM) and Tandem are able to exploit out-of-language data through posterior features, estimated by a Multi-Layer Perceptron (MLP). In this paper, we investigate the performance of posterior based approaches in the context of under-resourced speech recognition when a standard three-layer MLP is replaced by a deeper five-layer MLP. The deeper MLP architecture yields similar gains of about 15% (relative) for Tandem, KL-HMM as well as for a hybrid HMM/MLP system that directly uses the posterior estimates as emission probabilities. The best performing system, a bilingual KL-HMM based on a deep MLP, jointly trained on Afrikaans and Dutch data, performs 13% better than a hybrid system using the same bilingual MLP and 26% better than a subspace Gaussian mixture system only trained on Afrikaans data.

Index Terms— KL-HMM, Tandem, hybrid system, deep MLPs, under-resourced speech recognition

1. INTRODUCTION

Under-resourced speech recognition is a very challenging task. The main reason for this is the large amount of data that is usually required to train current recognizers. Therefore, acoustic modeling techniques that are able to exploit out-of-language data such as Kullback–Leibler divergence based HMM (KL-HMM) [1], Tandem [2] or Subspace Gaussian mixture models (SGMMs) [3] have been developed and extensively studied. KL-HMM and Tandem both exploit out-of-language data through posterior features, estimated by a Multi-Layer Perceptron (MLP) that was trained on out-of-language data. SGMMs on the other hand exploit out-of-language data through parameter sharing.

Recently, it has been shown that deep MLP architectures can greatly improve the performance of automatic speech recognition (ASR) systems [4]. Most deep MLP based ASR studies use hybrid HMM/MLP systems, where the MLP output is directly used to model the emission probability of the HMM states. However, if the MLP output is used as a feature [5, 6], conclusions tend to be more ambiguous, i.e. it is not clear if deeper MLP architectures are beneficial.

In this study, we build on our previous results [1] and investigate how deep MLP architectures affect the performance of posterior based acoustic modeling techniques that are particularly well suited for under-resourced ASR. As an additional reference point, we also evaluate SGMMs that do not rely on posterior features.

Taking Afrikaans as a representative of an under-resourced language (target language), we use large amounts of out-of-language data to improve an Afrikaans speech recognizer. Since Afrikaans is similar to Dutch, we intuitively expect that Dutch data provides most benefit for an Afrikaans speech recognizer [7]. Indeed, we already compared how English, Dutch and Swiss German data influence the performance of an Afrikaans speech recognizer and found that Dutch data yielded most improvement [8]. Hence, in this paper, we will use Dutch as a representative of the well-resourced language. In this context, we already compared phoneme accuracies of KL-HMM, Tandem, SGMM, conventional maximum likelihood linear regression (MLLR) and maximum a posteriori (MAP) adaptation systems [1]. Here, we compare word error rates (WERs) of KL-HMM, Tandem, SGMM and hybrid HMM/MLP systems. For KL-HMM, Tandem and HMM/MLP, we also investigate the impact of a deep MLP compared to the standard MLP.

The remainder of this paper is structured as follows: Section 2 described the databases that are used in this work. Section 3 then introduces all the investigated systems and Section 4 presents the experimental results.

The work was supported by Samsung Electronics Co. Ltd, South Korea, under the project “Multi-Lingual and Cross-Lingual Adaptation for Automatic Speech Recognition”, and by Eurostars Programme powered by Eureka and the European Community under the project “D-Box: A generic dialog box for multilingual conversational applications”.

ID	Language	number of	Amount of	
		phonemes	trn data	test data
AF	Afrikaans	38	3 h	50 min
CGN	Dutch	47	81 h	-

Table 1. Summary of the different languages with number of phonemes and amount of available data.

2. DATABASES

We used data from Afrikaans and Dutch as summarized in Table 1. In this section, we describe the two databases.

2.1. LWAZI

The Afrikaans data is available from the LWAZI corpus provided by the Meraka Institute, CSIR, South Africa[1] and described by [9]. The database consists of 200 speakers, recorded over a telephone channel at 8 kHz. Each speaker produced approximately 30 utterances, where 16 were randomly selected from a phonetically balanced corpus and the remainder consisted of short words and phrases.

The Afrikaans database comes with a dictionary [10] that defines the phoneme set containing 38 phonemes (including silence). The dictionary that we used contained 1,585 different words. The HLT group at Meraka provided us with the training and test sets. In total, about 3 h of training data and 50 min of test data is available (after voice activity detection).

The bi-gram language model, built on the training sentences, has 1.1% out-of-vocabulary words and a perplexity of about 19 on the test set.

2.2. Corpus Gesproken Nederlands

We used data of the Spoken Dutch Corpus (Corpus Gesproken Nederlands, CGN) [11] that contains standard Dutch pronounced by more than 4,000 speakers from the Netherlands and Flanders. The database is divided into several subsets and we only used *Corpus o* because it contains phonetically aligned *read* speech data pronounced by 324 speakers from the Netherlands and 150 speakers from Flanders. *Corpus o* uses 47 phonemes and contains 81 h of data after the deletion of silence segments that are longer than one second. It was recorded at 16 kHz, but since we use the data to perform ASR on Afrikaans, we downsampled it to 8 kHz prior to feature extraction.

3. SYSTEMS

In this section, we describe the systems under investigation. The systems can be divided into three different categories: (a) monolingual systems, using only Afrikaans data; (b) crosslingual systems, using only Dutch data during MLP training; and (c) bilingual systems, using Afrikaans and Dutch data during MLP training. This, coupled with the various different architectures, leads to quite a lot of systems. For a summary, see Table 5.

Afrikaans	HL	HU	OU	TRN	DEV
Standard	1	1,366	1,447	35.0%	30.8%
Deep	3	6,636	1,447	41.8%	35.0%

Table 2. Summary of the Afrikaans MLP training. The number of hidden layers (HL), the total number of hidden units (HU) and the number of output units (OU) are given. Frame accuracies on the training (TRN) and cross-validation set (DEV) are shown as well. Note that we fixed the number of hidden units to be the same than for the Dutch MLPs presented in Section 3.2.

3.1. Monolingual systems

The monolingual systems serve as reference systems only. In this paper, we evaluate a conventional HMM/GMM system, an SGMM system and two hybrid HMM/MLP systems, one based on a three-layer MLP (standard hybrid system) and one based on a five-layer MLP (deep hybrid system).

3.1.1. HMM/GMM

The HMM/GMM system is a standard cross-word context-dependent speech recognizer that models each triphone with three states and is based on 39 Mel-Frequency Perceptual Linear Prediction (MF-PLP) features (C_0–$C_{12} + \Delta + \Delta\Delta$), extracted with the HTK toolkit [12]. As usually done, we first trained context-independent monophone models that were then used as seed models for the context-dependent triphone models. We used eight Gaussians per state to model the emission probabilities. To balance the number of parameters with the amount of available training data, we applied conventional state tying with a decision tree that is based on the minimum description length principle [13], resulting in 1,447 tied states.

3.1.2. Monolingual SGMM

The SGMM acoustic modeling technique allows compact representation of large collection of mixture-of-Gaussian models and has shown its capability to outperform conventional HMM/GMMs in monolingual as well as cross- or multi-lingual scenarios [3, 14]. For the monolingual SGMM system, we trained all the parameters from Mel-Frequency cepstrum coefficients (MFCCs), using Afrikaans data only. In total we used 500 Gaussians and the substate phone-specific vectors had 50 dimensions.

[1] http://www.meraka.org.za/hlt

Dutch	HL	HU	OU	TRN	DEV
Standard	1	1,366	1,789	59.0%	56.5%
Deep	3	6,636	1,789	64.2%	60.3%

Table 3. Summary of the Dutch MLP training. The number of hidden layers (HL), the total number of hidden units (HU) and the number of output units (OU) are given. Frame accuracies on the training (TRN) and cross-validation set (DEV) are shown as well.

3.1.3. Monolingual HMM/MLP

The monolingual HMM/MLP systems used the same 1,447 tied states as the HMM/GMM system presented in Section 3.1.1. For the standard hybrid system, we trained a three-layer MLP and for the deep hybrid system, we trained a five-layer MLP (each hidden layer had similar number of hidden units) using Quicknet software [15]. We randomly split the three hours of Afrikaans training data into an *MLP training set* (90%) and an *MLP cross-validation set* (10%). We trained the MLPs from the 39-dimensional MF-PLP features in a nine frame temporal context (four preceding and following frames). More details about the MLP training are given in Table 2. For this study, the only difference between the three-layer and the five-layer network was in the number of parameters (and in the number of hidden layers). We did not employ more elaborated training procedures such as pre-training or dropout. The resulting posterior probabilities were divided by the priors and then directly used as emission probabilities.

3.2. Crosslingual systems

The crosslingual systems exploit Dutch data during MLP training. More specifically, we trained a standard and a deep MLP with all the available Dutch data. As we already did in earlier studies [1], we developed a standard HMM/GMM system with all the Dutch training data to obtain 1,987 tied states targets. We set the number of parameters for the standard MLP to 10% of the available number of training frames, resulting in a hidden layer with 1,366 units. As suggested by studies on deep MLPs [16], we targeted about 2000 hidden units per layer in the deeper MLP and therefore set the number of parameters to 50% of the available number of training frames, leading to a total of 6,636 hidden units distributed to three hidden layers. We used 90% of the training set for MLP training and 10% for cross-validation to stop training. More details about the MLP training are given in Table 3.

In this paper, we investigated two approaches that benefit from exploiting out-of-language data through posterior features: Tandem and KL-HMM. In both approaches, Afrikaans data is passed through the MLP trained on Dutch and the resulting posterior features are then used to train the HMM parameters (see Sections 3.2.1 and 3.2.2). Since the hybrid HMM/MLP approach is bound to the tied states target used during the MLP training, we did not evaluate a crosslingual HMM/MLP system. However, as an additional reference point, we also evaluated a crosslingual SGMM system that did not use the Dutch posterior features, but used the Dutch data for global parameter training as described in [1].

3.2.1. Crosslingual Tandem

Similar to the conventional HMM/GMM system, for the Tandem system, we trained context-independent monophone models that served as seed models for the three-state context-dependent triphone models. Because of the ambiguous results from earlier studies [5, 6], we evaluate a *standalone Tandem* system (similar to the system in [6]) as well as an *augmented Tandem* system, where *augmented* refers to our concatenating of MF-PLP features with the posterior features (similar to the system in [5]). We used eight Gaussians per state to model the emission probabilities. As in our previous study [1], we used PCA for dimensionality reduction and fixed the dimensionality such that 99% of the variance was preserved. This procedure resulted in 286-dimensional features (we used the same feature dimensionality for the posteriors of the standard and the deep MLP). To have comparable Tandem systems, we run PCA again after concatenating MF-PLP features with posterior features and reduced the dimensionality to 286.

3.2.2. Crosslingual KL-HMM

The KL-HMM acoustic modeling technique can directly model raw posterior features. Therefore no post-processing is necessary. In the KL-HMM acoustic modeling approach, the HMM states are parametrized with reference posterior distributions (categorical distributions) that can be trained by minimizing the Kullback–Leibler divergence between the categorical distributions and the posterior features. More details about training and decoding in the KL-HMM framework can be found in, for instance, [1]. Similar to HMM/GMM and Tandem, the KL-HMM system was trained based on the context-independent monophone models that served as seed models for the three-state context-dependent triphone models. For KL-HMM, we applied a decision tree clustering reformulated as dictated by the KL criterion [17]. We found in our previous study that the best KL-HMM performance is achieved with a fully developed tree (about 15,000 tied states), therefore we did the same for this study.

3.2.3. Crosslingual SGMM

SGMMs can be naturally exploited in under-resourced scenarios, since most of the model parameters can be estimated on well-resourced datasets. Therefore, we use the crosslingual SGMM system as an additional reference point in this study. To exploit out-of-language data, the SGMM model parameters can be divided into HMM-state specific and shared

AF & Dutch	HL	HU	OU	TRN	DEV
Standard	1	1,366	1,447	48.3%	38.3%
Deep	3	6,636	1,447	53.1%	42.1%

Table 4. Summary of the MLP trained on Dutch first and the re-trained on Afrikaans. The number of hidden layers (HL), the total number of hidden units (HU) and the number of output units (OU) are given. Frame accuracies on the training (TRN) and cross-validation set (DEV) are shown as well.

parameters. The crosslingual SGMM used Dutch data during training of the globally-shared (language-independent) parameters and Afrikaans data for the training of the HMM-state specific parameters [3]. Similar to the monolingual SGMM system, we used 500 Gaussians and the substate phone-specific vectors had 50 dimensions.

3.3. Bilingual systems

Inspired by a recent study [6], the bilingual systems that we present are based on MLPs that were trained on Afrikaans and Dutch data. More specifically, we took the two Dutch MLPs (standard and deep) trained in Section 3.2 and removed the output layer. Then, we appended a new randomly initialized output layer and trained the MLP (all layers) to estimate posterior probabilities for the 1,447 Afrikaans tied states by using Afrikaans data. More details about the MLP training are given in Table 4. In this study, we investigated three acoustic modeling techniques that are able to exploit the posterior probabilities estimated with the bilingually trained MLP: hybrid HMM/MLP, Tandem and KL-HMM. Again, SGMM serves as a reference not using posterior features.

3.3.1. Bilingual HMM/MLP

The bilingual HMM/MLP systems are essentially the same systems as the monolingual HMM/MLP ones presented in Section 3.1.3. The monolingual HMM/MLP systems used the posterior probabilities estimated with the MLP only trained on Afrikaans data, and the bilingual HMM/MLP systems employed the posterior probabilities estimated with the MLP first trained on Dutch data and then re-trained on Afrikaans data.

3.3.2. Bilingual Tandem

Similar to the crosslingual Tandem systems, presented in Section 3.2.1, we trained a standalone and an augmented Tandem system based on three-state context-dependent triphone models. We used eight Gaussians per state to model the emission probabilities and used PCA for decorrelation. To preserve 99% of the variance we reduced the feature dimensionality to 146.

3.3.3. Bilingual KL-HMM

The bilingual KL-HMM system resembles the crosslingual KL-HMM system, presented in Section 3.2.2. The 1,789 dimensional Dutch posterior features were replaced by 1,447 dimensional feature vectors, trained on Dutch and on Afirkaans data.

3.3.4. Bilingual SGMM

The bilingual SGMM system used Dutch and Afrikaans data during training of the globally-shared parameters and Afrikaans data only for the training of the HMM-state specific parameters. We used 500 Gaussians and the substate phone-specific vectors had 50 dimensions.

4. EXPERIMENTS

In this section, we first discuss the hypotheses under investigation, then present the experimental results.

4.1. Prior expectations

Given the systems described in Section 3, we hypothesize the following:

1. Based on the success of deep architectures in recent studies [4], we hypothesize that the deep MLP architectures yield improvement for all systems.

2. Recent literature [5] suggests that adding hidden layers does not improve the performance of a augmented Tandem system. We therefore assume that MLP output post-processing reduces the performance gain resulting from deeper MLP architectures and hypothesize that:

   (a) hybrid systems gain most from a deeper MLP architecture because they directly use the estimated posteriors probabilities as emission probabilities.

   (b) KL-HMM gains more than Tandem because the posterior features are directly modeled without post-processing.

3. Multilingual data was successfully used to generate deep neural network features for low resource speech recognition [6]. Therefore, we hypothesize that the gains from the deep MLP architecture and the out-of-language data exploitation are complementary.

4.2. Results

The experimental results are summarized in Table 5. All the systems based on deep MLPs outperform the equivalent system based on the standard MLP, hence hypothesis 1 is demonstrated.

	System	Std.	Deep	Rel. Gain
Monoling.	HMM/GMM	11.4%	-	-
	SGMM	9.5%	-	-
	HMM/MLP	12.3%	9.9%	20%
Crossling.	Tandem	10.5%	9.4%	10%
	+MF-PLP	9.7%	9.5%	2%
	KL-HMM	9.6%	9.0%	6%
	SGMM	8.5%	-	-
Biling.	HMM/MLP	9.3%	8.0%	14%
	Tandem	9.9%	8.4%	15%
	+MF-PLP	9.7%	8.9%	8%
	KL-HMM	8.0%	7.0%	13%
	SGMM	8.5%	-	-

Table 5. Achieved word error rates (WERs) of the monolingual, crosslingual and bilingual systems described in Section 3. *Std.* stands for the standard (three-layer) MLP and *deep* for the deep (five-layer) MLP. The relative gain by using the deeper MLP is also given.

For the bilingual scenario, HMM/MLP, KL-HMM and standalone Tandem yield very similar improvement if the standard and deep MLP performance are compared. Therefore we must reject hypothesis 2. We evaluated a standalone and an augmented Tandem system. Our results are in line with earlier studies [5, 6] where it was found that deep MLPs yield improvement for standalone systems [6], but only to a limited extend for augmented Tandem systems [5]. It seems reasonable to conclude that the concatenation of the MLP output with MF-PLP features diminishes the advantage of the deep MLP architecture.

Although the experimental results suggest that the relative gain decreases in cross- and bi-lingual scenarios compared to the monolingual HMM/MLP system, it seems that the gains from out-of-language data exploitation and a deep MLP architecture are still complementary. Thus, hypothesis 4 is demonstrated.

The bilingual KL-HMM systems yields the best performance (13% relative improvement compared to the hybrid HMM/MLP system). We attribute the advantage of the KL-HMM system to the fact that the hybrid system is bound to the tied state targets used during the MLP training. Hence the hybrid system uses about 1,500 tied states. The KL-HMM system on the other hand is more flexible and allows more tied states to be used, even in under-resourced scenarios. The parsimonious use of parameters of the KL-HMM system (categorical distributions) allows training of an HMM with 15,000 tied states, only using three hours of Afrikaans data.

Furthermore, Table 5 also reveals that the crosslingual and the bilingual SGMM perform similarly. The crosslingual environment is particularly well suited for the SGMM system because the shared parameters can be trained on Dutch data and the language specific parameters on Afrikaans data. In the bilingual case however, the 3 h of Afrikaans data are dominated by the 80 h of Dutch data during the shared parameter training. The MLP based systems yield more improvement from the bilingual setup because the MLPs estimate Afrikaans tied states posteriors instead of Dutch tied states posteriors in the crosslingual case.

5. CONCLUSION

We investigated under-resourced speech recognition in the context of an Afrikaans speech recognizer that benefits from Dutch data, and compared how the performance of posterior based approaches changes if a standard three-layer MLP is replaced by a deeper five-layer MLP. We have shown that the deeper MLP structure equally improved a hybrid HMM/MLP and a standalone Tandem system as well as a KL-HMM system. Further, experiments revealed that gains from the deeper MLP architecture and out-of-language data exploitation are complementary. The best performing bilingual system, KL-HMM based on the MLP that was jointly trained on Afrikaans and Dutch data, performs 13% better than a hybrid system using the same bilingual MLP and yields 26% relative improvement if compared to a monolingual SGMM system only trained on Afrikaans data.

We therefore conclude that deep MLP architectures are suitable for under-resourced speech recognition, with the KL-HMM being the most promising.

6. REFERENCES

[1] D. Imseng, P. Motlicek, H. Bourlard, and P. N. Garner, “Using out-of-language data to improve an under-resourced speech recognizer,” *Speech Communication*, 2013.

[2] A. Stolcke, F. Grezl, M.-Y. Hwang, X. Lei, N. Morgan, and D. Vergyri, “Cross-domain and cross-language portability of acoustic features estimated by multilayer perceptrons,” in *Proc. of ICASSP*, 2006, vol. 1, pp. 321–324.

[3] L. Burget, P. Schwarz, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N. Goel, M. Karafiát, D. Povey, A. Rastrow, R. C. Rose, and S. Thomas, “Multilingual acoustic modeling for speech recognition based on subspace gaussian mixture models,” in *Proc. of ICASSP*, 2010, pp. 4334–4337.

[4] G. Hinton, Li Deng, Dong Yu, G.E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, and B. Kingsbury, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[5] P. Swietojanski, A. Ghoshal, and S. Renals, "Unsupervised cross-lingual knowledge transfer in DNN-based LVCSR," in *Proc. of the IEEE Workshop on Spoken Language Technology (SLT)*, 2012, pp. 246–251.

[6] S Thomas, M. L. Seltzer, Church K., and Hermansky H., "Deep neural network features and semi-supervised training for low resource speech recognition," in *Proc. of ICASSP*, 2013, pp. 6704–6708.

[7] W. Heeringa and F. de Wet, "The origin of Afrikaans pronunciation: a comparison to west Germanic languages and Dutch dialects," in *Proceedings of the Conference of the Pattern Recognition Association of South Africa*, 2008, pp. 159–164, `www.let.rug.nl/heeringa/dialectology/papers/prasa08.pdf`.

[8] D. Imseng, H. Bourlard, and P. N. Garner, "Boosting under-resourced speech recognizers by exploiting out of language data - case study on Afrikaans," in *Proceedings of the 3rd International Workshop on Spoken Languages Technologies for Under-resourced Languages*, 2012, pp. 60–67.

[9] E. Barnard, M. Davel, and C. van Heerden, "ASR corpus design for resource-scarce languages," in *Proc. of Interspeech*, 2009, pp. 2847–2850.

[10] M. Davel and O. Martirosian, "Pronunciation dictionary development in resource-scarce environments," in *Proc. of Interspeech*, 2009, pp. 2851–2854.

[11] N. Oostdijk, "The spoken Dutch corpus. Overview and first evaluation.," in *In Proceedings of the Second International Conference on Language Resources and Evaluation*, 2000, vol. II, pp. 887–894.

[12] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland, *The HTK Book, version 3.4*, Cambridge University Engineering Department, Cambridge, UK, 2006.

[13] K. Shinoda and T. Watanabe, "Acoustic modeling based on the MDL principle for speech recognition," in *Proc. of Eurospeech*, 1997, pp. 99–102.

[14] P. Motlicek, D. Povey, and M. Karafiat, "Feature and score level combination of subspace Gaussians in LVCSR task," in *Proc. of ICASSP*, 2013, pp. 7604–7608.

[15] D. Johnson, "ICSI quicknet software package," `http://www.icsi.berkeley.edu/Speech/qn.html`, 2004.

[16] Abdel-rahman Mohamed, George E Dahl, and Geoffrey Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.

[17] D. Imseng, J. Dines, P. Motlicek, P. N. Garner, and H. Bourlard, "Comparing different acoustic modeling techniques for multilingual boosting," in *Proc. of Interspeech*, 2012.

5.3 Paper 3: [Mad+20a]

Lattice-Free Maximum Mutual Information Training of Multilingual Speech Recognition Systems

Srikanth Madikeri[1], *Banriskhem K. Khonglah*[1], *Sibo Tong* [1,2], *Petr Motlicek*[1], *Hervé Bourlard*[1,2], *Daniel Povey*[3]

[1]Idiap Research Institute, Martigny, Switzerland
[2]Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland
[3]Xiaomi Technology, China

msrikanth, bkhonglah, stong, petr.motlicek, bourlard@idiap.ch, dpovey@xiaomi.com

Abstract

Multilingual acoustic model training combines data from multiple languages to train an automatic speech recognition system. Such a system is beneficial when training data for a target language is limited. Lattice-Free Maximum Mutual Information (LF-MMI) training performs sequence discrimination by introducing competing hypotheses through a denominator graph in the cost function. The standard approach to train a multilingual model with LF-MMI is to combine the acoustic units from all languages and use a common denominator graph. The resulting model is either used as a feature extractor to train an acoustic model for the target language or directly fine-tuned. In this work, we propose a scalable approach to train the multilingual acoustic model using a typical multitask network for the LF-MMI framework. A set of language-dependent denominator graphs is used to compute the cost function. The proposed approach is evaluated under typical multilingual ASR tasks using GlobalPhone and BABEL datasets. Relative improvements up to 13.2% in WER are obtained when compared to the corresponding monolingual LF-MMI baselines. The implementation is made available as a part of the Kaldi speech recognition toolkit.

Index Terms: speech recognition, multilingual ASR, LF-MMI

1. Introduction

In Automatic Speech Recognition (ASR) for low-resourced languages, training multilingual systems is an effective way to compensate for limited amount of data [1, 2, 3, 4, 5, 6]. When trained with resources from multiple languages, Deep Neural Networks (DNN) based Acoustic Models (AM) can function as a feature extractor to train a monolingual acoustic model for the target language [7, 8, 9]. Alternately, the models can be adapted to the target language [10, 11, 12, 13, 14, 15]. The multilingual models can either share the output layer or have separate output layers (one for each language) [3]. In the former case, monophones may be used to avoid a huge output layer, which is often followed by retraining the network for the target language with senones.

In this work, we focus on sequence-discriminative training of multilingual AM with the Lattice-Free Maximum Mutual Information (LF-MMI) framework [16]. LF-MMI training has been shown to have superior performance compared to the conventional cross-entropy (CE) training of DNNs [17, 18]. The MMI cost function uses a numerator graph modelling the observed feature sequence based on ground truth and a denominator graph computing the probability over all possible sequences [19]. The latter enforces the discriminative property in the training shown to be useful for training AM [20, 21].

Given the advantages of both multilingual and LF-MMI training procedures, it is natural to combine them to obtain better performance. In [22, 23], multilingual LF-MMI models were observed to improve over their monolingual counterparts. The multilingual resources are combined by merging the phoneme sets from all languages either using a universal phone set such as the International Phonetic Alphabet (IPA), or by combining acoustic units. In both cases, a universal denominator graph is shared across all languages during training.

When combining acoustic units for multilingual training, the output layer size increases rapidly with number of languages. This may render such a system impractical during decoding. We refer to this type of multilingual AM as a single-task system. Alternately, multitask training solves this issue by separating the output layers of languages so that during decoding only the output relevant to the language is used. An added advantage during training is that the cost function can be computed faster as its complexity depends on the number of states in the denominator.

In this paper, we compare different styles of multilingual training in the LF-MMI framework. The two styles are broadly categorized as single-task and multitask depending on whether the output layer is shared across languages or not. For single-task training, existing LF-MMI implementation can be easily extended. For multitask training, we make our implementation available as a part of Kaldi [24][1]. The comparisons are performed on two commonly used multilingual databases: (1) GlobalPhone and (2) BABEL. We present results on 5 target languages for the former and 4 target languages for the latter. The results show that multitask training provides a much more scalable approach to develop multilingual AM due to the aforementioned advantages without any loss in performance. The rest of the paper is organized as follows: in Section 2, the LF-MMI training procedure is described. In Section 3, the proposed multilingual LF-MMI training procedure is given. Results of our experiments on GlobalPhone and BABEL are described in Section 4.

The research is based upon the work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via AFRL Contract #FA8650-17-C-9116. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The work is also partially supported by the ROXANNE H2020 EC project (http://www.roxanne-euproject.org), under grant agreement No. 833635.

[1] egs/babel_multilang/s5d/local/chain2/run_tdnn.sh

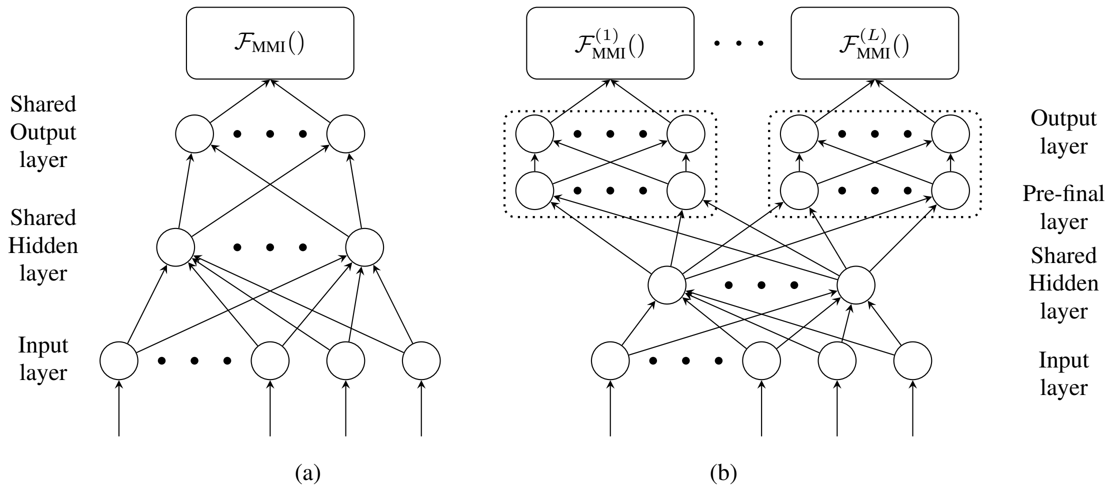


Figure 1: *(a) Multilingual LF-MMI system with shared output layer. The objective function $\mathcal{F}_{MMI}$ is computed with either a language-independent or language-dependent denominator graphs. (b) Proposed LF-MMI system with language-dependent objective functions. Both systems are shown for a simple feedforward architecture.*

2. LF-MMI

In LF-MMI, the MMI objective function is used as the cost function to train the AM [16]. The cost function is given as follows:

$$\mathcal{F}_{\text{MMI}} = \sum_{u=1}^{U} \log \frac{p\left(\mathbf{x}^{(u)}|\mathcal{M}_{\mathbf{w}(u)}, \boldsymbol{\theta}\right) p(\mathbf{w}(u))}{p\left(\mathbf{x}^{(u)}|\mathcal{M}_{\text{den}}, \boldsymbol{\theta}\right)}, \qquad (1)$$

where $\mathbf{x}^{(u)}$ is the input sequence,

u is an utterance,

U is the set of all training utterances,

$\mathcal{M}_{\mathbf{w}(u)}$ corresponds to the numerator graph specific to a word sequence in transcription,

$\mathcal{M}_{\text{den}}$ is the denominator graph modelling all possible word sequences and

$\boldsymbol{\theta}$ is the model parameter.

The numerator can be computed either using alignments from another acoustic model, or in a completely end-to-end fashion [17]. In this work, we always use alignments from a monolingual HMM/GMM model.

The standard implementation of LF-MMI makes several simplifications to the conventional AM training of DNNs. First, the HMM topology is modified to a 2-state HMM so that the final state can be reached in one frame. Next, frame-dropping is employed during training so that only 1 in 3 frames is required during decoding. Finally, the segment length of an utterance during training is limited.

The derivatives of the two quantities–numerator and denominator–in Equation 1 are computed using two graphs. The numerator graph is constructed using forced alignment and the denominator graph is obtained by composing the phone language model with the phonetic context-dependency followed by context-dependent states. Following the notation in [18], if ${}^{\text{NUM}}\gamma_t^{(u)}(s)$ is the posterior from the numerator at time t for state s and ${}^{\text{DEN}}\gamma_t^{(u)}(s)$ is that from the denominator, the gradient is given by:

$$\frac{\partial \mathcal{F}_{\text{MMI}}}{\partial y_t^{(u)}(s)} =^{\text{NUM}} \gamma_t^{(u)}(s) -^{\text{DEN}} \gamma_t^{(u)}(s), \qquad (2)$$

where $y_t^{(u)}(s)$ is the network output for state s at time t given input utterance u

While training a multilingual model with the output layer containing acoustic units from all languages, the objective still remains the same as above, meaning the ${}^{\text{DEN}}\gamma$ is language-independent.

Computing ${}^{\text{DEN}}\gamma$ requires training a phone language model. Combining acoustic units across all languages for single-task multilingual training not only increases the number of states in the denominator graph, but may also introduce lead to noisy ${}^{\text{DEN}}\gamma$ estimates. Thus, to reduce the influence of other languages while computing ${}^{\text{DEN}}\gamma$, we propose to use a set of language-dependent denominator for AMs trained in multitask fashion.

3. Multilingual LF-MMI

Multiple approaches exist to train a multilingual AM. Depending on whether the output layer is shared by languages or not, we can classify it as either single-task or multitask model. The difference between these two broad categories of multilingual LF-MMI systems is shown in Figure 1. In the multitask architecture, each language has a separate output layer preceded by a pre-final layer and a corresponding objective function (marked $\mathcal{F}^{(1)}_{\text{MMI}}, \ldots, \mathcal{F}^{(L)}_{\text{MMI}}$ in the figure).

The choice between single-task and multitask AM dictates how the acoustic units are shared across languages. In the single-task case, one can simply combine the acoustic units by choosing a union of all non-silence acoustic units from each language. Alternately, well-defined linguistic units such as IPA can be used to derive the acoustic units. In the multitask case, each language will have its own set of acoustic units.

Given such possibilities to train the AM, the single-task configuration also provides a choice of using language-specific (i.e. trained with data from all languages) or language-independent denominator (i.e. trained with data from only one language), whereas only the language-independent denominator is applicable in the multitask case. The focus of this paper is to compare all such possible configurations to better understand the performance of the resulting models.

In single-task multilingual AM, the case of using language-independent denominator is equivalent to training monolingual AMs. However, when using language-specific denominators, the cost function changes as follows: we have L objective functions, where L is the number of languages, computed independent of each other depending only on the language of the utterance:

$$\mathcal{F}_{\text{MMI}}^{(\ell)} = \sum_{u=1}^{U_\ell} \log \frac{p\left(\mathbf{x}^{(u)} | \mathcal{M}_{\mathbf{w}(u)}^{\ell}, \boldsymbol{\theta}\right) p(\mathbf{w}(u))}{p\left(\mathbf{x}^{(u)} | \mathcal{M}_{\text{den}}^{\ell}, \boldsymbol{\theta}\right)}, \tag{3}$$

where U_ℓ is the number of utterances in the minibatch for language ℓ, $\boldsymbol{\theta}$ contains the shared and language-dependent parameters, $\mathcal{M}_{\mathbf{w}(u)}^{\ell}$ and $\mathcal{M}_{\text{den}}^{\ell}$ are language-specific numerator and denominator graphs, respectively.

Each denominator graph is built from the language-specific phone language model (the same as that used in monolingual LF-MMI training). Gradients for language-dependent layers are computed and updated for each minibatch. Using backpropogation, the shared parameters are then updated. The overall cost-function is the weighted sum of all language-dependent cost-functions:

$$\mathcal{F}_{\text{MMI}} = \sum_{\ell=1}^{L} \alpha_\ell F_{MMI}^{\ell}, \tag{4}$$

where α_ℓ is language-dependent weight. Note that each minibatch is expected to have samples (sequence of MFCCs) from multiple languages. To facilitate such a training in Kaldi, we modify the training procedure to select the denominator graph for each sequence in the minibatch according to the language. In practice, this only requires the knowledge of the language of each sequence in the minibatch. Assuming the sequences are grouped by language, we simply iterate over the languages in the minibatch to call the existing procedures for monolingual training with the appropriate denominator graph. Such multitask models also simplify addition or removal of languages and applying language-specific operations during training.

4. Experiments

Experiment results are reported on GlobalPhone [25] and BABEL datasets. All experiments are performed with the Kaldi toolkit [24]. For GlobalPhone, we used the French (FR), German (GE), Portuguese (PO), Russian (RU) and Spanish (SP) datasets from the GlobalPhone corpus [23]. Each language has roughly 20 hours of speech for training and two hours for development and evaluation sets, from a total of about 100 speakers. The development sets were used to tune the hyper-parameters for training. Only the results on evaluation sets are reported. The trigram language models that we used are publicly available[2]. The detailed statistics for each of the languages is given in Table 1.

We also investigated our proposed method with the BABEL dataset. Datasets for several languages with limited resources were released during the BABEL project with the main goal of building keyword spotting systems. We considered 4 BABEL languages for evaluation: Tagalog (TGL), Swahili (SWA), Zulu (ZUL), and Turkish (TUR). The statistics of the target languages are given in Table 2. Trigram language models are used during testing.

Table 1: *Statistics of the subset of GlobalPhone languages used in this work: the amounts of speech data for training and evaluation sets are in hours.*

Language	Vocab	PPL	#Phones	Train	Dev	Eval
FR	65k	324	38	22.7	2.1	2.0
GE	38k	672	41	14.9	2.0	1.5
PO	62k	58	45	22.7	1.6	1.8
RU	293k	1310	48	21.1	2.7	2.4
SP	19k	154	40	17.6	2.0	1.7

Table 2: *Statistics of BABEL target languages used for testing. Note that the Eval sets mentioned refer to the "dev" set in the official BABEL release. Only conversational speech is considered for both training and testing. All durations are calculated prior to silence removal. (PPL: perplexity)*

Language	Vocabulary	PPL	Train (h)	Eval (h)
Tagalog	22k	148	84.5	10.7
Swahili	25k	357	38.0	9.3
Turkish	41k	396	77.2	9.8
Zulu	56k	719	56.7	9.2

4.1. GlobalPhone Setup

We used 40-dimensional MFCCs as acoustic features, derived from 25 ms frames with a 10 ms frame shift. The features were normalized via mean subtraction and variance normalization on a speaker basis. We used a frame subsampling factor of 3 which speeds up training by a factor of 2. We also augmented the data with 2-fold speed perturbation in all the experiments. The network consists of 8 layers of Time Delay Neural Network (TDNN), with 450 nodes in each layer [26].

We compare the monolingual systems to three multilingual systems: (1) single-task system trained with language independent denominator, (2) single-task system trained with language dependent denominator, and (3) multitask system trained with language dependent denominator. For the single-task systems, we concatenate the phonemes from the five languages to create the universal phone set for multilingual training. We did not use IPA-based phone set as in [23] because we found that the concatenated phone set performs better in preliminary experiments.

4.2. GlobalPhone Results

The results on GlobalPhone are presented in Table 3. The single-task multilingual systems trained with concatenated phone set improve over the monolingual LF-MMI systems on four out of five languages. Using language-dependent denominator, in this case, does not make a significant difference in terms of WERs, thus only providing computational benefits during training. The single-task system performs better on FR and GE than the multitask system. The difference on the other languages is marginal. The multitask multilingual system improves over the monolingual baseline for 4 out of 5 languages. The relative improvements range from 0.7% (for PO) to 10% (for RU). We do not compare to the CE system as its results are poorer compared to the two LF-MMI baselines. We believe that the LF-MMI baselines are superior due to the controlled nature of the dataset (read speech and clean acoustic conditions).

[2] http://www.csl.uni-bremen.de/GlobalPhone/

Table 3: *Comparison between target languages in GlobalPhone in WER(%). (FR: French, GE: German, PO: Portugese, RU: Russian, SP: Spanish)*

System	FR	GE	PO	RU	SP
Monolingual LF-MMI	20.4	12.7	15.2	24.6	7.1
Single-task multilingual system					
Language independent	21.3	12.5	**14.9**	**22.1**	6.6
Language dependent	21.3	12.4	15.0	**22.1**	6.6
Multitask multilingual system					
5 languages	20.7	**11.7**	15.1	**22.1**	**6.5**

Table 4: *BABEL languages used for training and testing.*

Category	Languages
Target languages & 4 Language Training	Tagalog, Swahili, Zulu, Turkish
14 Language Training	Tagalog, Swahili, Zulu, Turkish, Assamese, Bengali, Cantonese, Haitian, Kazhak, Kurmanji, Tamil, Telugu, Tok, Vietnamese

4.3. BABEL setup

We consider two training configurations: training with only 4 of the target languages and training with 14 languages. The 14-language system is used to demonstrate the scalability of the multitask system. In both cases, results for only 4 target languages are reported (see Table 4). We follow the feature configuration (except for feature mean and variance normalization) and data augmentation of GlobalPhone systems. In addition, an online i-vector extractor of dimension 100 is trained for each configuration. The transcripts are used for speech/non-speech labels. The online i-vectors are appended to MFCCs as input to the DNN. TDNN architecture is used with 8 hidden layers. Each hidden layer has 1024 units. The pre-final layer has only 200 units. Frame-dropping is enabled for all models.

In order to obtain alignments to train all the TDNN models, HMM/GMM models were first trained for each language. The standard recipe from Kaldi was followed.

4.4. BABEL results

The results on target languages from BABEL are presented in Table 5. The performance of the monolingual LF-MMI models are already better compared to those presented in literature, thus forming a strong baseline. Next, we compare the monolingual models to three multilingual models trained with the 4 language setup: (1) single-task system trained with language independent denominator, (2) single-task system trained with language dependent denominator, and (3) multitask system trained with language dependent denominator. The results show that in conditions with high acoustic variability, as in the case of BABEL data-sets, multilingual training brings considerable benefits. The multilingual systems show improvements over the monolingual systems for all languages. This clearly demonstrates the benefit of multilingual LF-MMI training for low-resource languages. Both single-task and multitask setups outperform the monolingual baseline, with relative improvements ranging from 2% to 8.8% for the former and 3% to 8% for the latter. In the single-task setup, as in the case of Globalphone, language-dependent denominator provides only marginal gains over language-independent denominator. Overall, the benefits obtained are dependent on the language, but no significant loss is observed by choosing one technique for multilingual training over the other for majority of the languages (Zulu being the exception).

Table 5: *Comparison between target languages in BABEL in WER(%). Improvements with LF-MMI are in bold. (TGL: Tagalog, SWA: Swahili, TUR: Turkish, ZUL: Zulu)*

System	TGL	SWA	TUR	ZUL
Monolingual LF-MMI	45.3	38.7	47.2	53.5
Single-task multilingual system				
Language independent	44.4	35.5	43.4	52.4
Language dependent	44.4	**35.4**	**43.0**	51.9
Multitask multilingual system				
4 languages	**43.9**	35.6	43.5	**51.0**
14 languages	**42.2**	**33.6**	43.9	**50.8**

To demonstrate the scalability of the multitask system, we also train an AM with 14 languages (final row in Table 5; the 14 languages are in Table 4). Compared with the 4 languages system, the 14 language system improves on 3 out of 4 languages. Relative improvements range from 0.4% (ZUL) to 5.6% (SWA) suggesting that adding more languages to the AM training can be beneficial without any additional cost during decoding. In addition, compared to the monolingual baseline relative improvement of up to 13.2% (SWA) is obtained.

5. Conclusions

In this work, we compared different styles of training multilingual acoustic models in the LF-MMI framework. The system was evaluated on GlobalPhone and BABEL datasets. The results on target languages in GlobalPhone show that the multitask training approach leads to a system that outperforms single-task models trained with either IPA or combined phone sets. The results on BABEL datasets show similar trends in improvement for 3 out of 4 target languages. By further increasing the number of languages in training significant benefits are achieved demonstrating the scalability of our method. We obtained relative improvements up to 13.2% when compared to the monolingual model.

6. References

[1] N. T. Vu and T. Schultz, "Multilingual multilayer perceptron for rapid language adaptation between and across language families." in *Interspeech*, 2013, pp. 515–519.

[2] L. Burget *et al.*, "Multilingual acoustic modeling for speech recognition based on subspace gaussian mixture models," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2010, pp. 4334–4337.

[3] M. Karafiát *et al.*, "Multilingual blstm and speaker-specific vector adaptation in 2016 but babel system," in *2016 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 637–643.

[4] 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.

[5] D. Imseng, P. Motlicek, H. Bourlard, and P. N. Garner, "Using out-of-language data to improve an under-resourced speech recognizer," *Speech communication*, vol. 56, pp. 142–151, 2014.

[6] N. T. Vu, D. Imseng, D. Povey, P. Motlicek, T. Schultz, and H. Bourlard, "Multilingual deep neural network based acoustic modeling for rapid language adaptation," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 7639–7643.

[7] K. Vesel`y, M. Karafiát, F. Grézl, M. Janda, and E. Egorova, "The language-independent bottleneck features," in *2012 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2012, pp. 336–341.

[8] F. Grézl, M. Karafiát, and K. Vesel`y, "Adaptation of multilingual stacked bottle-neck neural network structure for new language," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 7654–7658.

[9] Z. Tüske, J. Pinto, D. Willett, and R. Schlüter, "Investigation on cross-and multilingual mlp features under matched and mismatched acoustical conditions," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 7349–7353.

[10] A. Ghoshal, P. Swietojanski, and S. Renals, "Multilingual training of deep neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 7319–7323.

[11] S. Tong, P. N. Garner, and H. Bourlard, "Cross-lingual adaptation of a ctc-based multilingual acoustic model," *Speech Communication*, vol. 104, pp. 39–46, 2018.

[12] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 7304–7308.

[13] D. Imseng, B. Potard, P. Motlicek, A. Nanchen, and H. Bourlard, "Exploiting un-transcribed foreign data for speech recognition in well-resourced languages," in *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2014, pp. 2322 – 2326.

[14] P. Motlicek, D. Imseng, B. Potard, P. N. Garner, and I. Himawan, "Exploiting foreign resources for dnn-based asr," *EURASIP Journal on Audio, Speech, and Music Processing*, no. 2015:17, Jun. 2015.

[15] A. Srinivasamurthy, P. Motlicek, I. Himawan, G. Szaszak, Y. Oualil, and H. Helmke, "Semi-supervised learning with semantic knowledge extraction for improved speech recognition in air traffic control," in *Proceedings of Interspeech 2017*, Aug. 2017, pp. 2406–2410.

[16] D. Povey *et al.*, "Purely sequence-trained neural networks for asr based on lattice-free mmi." in *Interspeech*, 2016, pp. 2751–2755.

[17] H. Hadian, H. Sameti, D. Povey, and S. Khudanpur, "End-to-end speech recognition using lattice-free mmi," in *Proc. Interspeech*, 2018, pp. 12–16.

[18] H. Hadian *et al.*, "Flat-start single-stage discriminatively trained hmm-based models for asr," *IEEE ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 11, pp. 1949–1961, 2018.

[19] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, University of Cambridge, 2005.

[20] K. Vesel`y, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks." in *Interspeech*, vol. 2013, 2013, pp. 2345–2349.

[21] B. Kingsbury, "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2009, pp. 3761–3764.

[22] F. Keith, W. Hartmann, M.-H. Siu, J. Ma, and O. Kimball, "Optimizing multilingual knowledge transfer for time-delay neural networks with low-rank factorization," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4924–4928.

[23] S. Tong, P. N. Garner, and H. Bourlard, "An investigation of multilingual asr using end-to-end lf-mmi," *In Proc. of ICASSP 2019*, pp. 6061–6065, 2019.

[24] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," IEEE Signal Processing Society, Tech. Rep., 2011.

[25] T. Schultz, N. T. Vu, and T. Schlippe, "Globalphone: A multilingual text & speech database in 20 languages," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 8126–8130.

[26] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

5.4 Paper 4: [MMB21]

Multitask adaptation with Lattice-Free MMI for multi-genre speech recognition of low resource languages

Srikanth Madikeri[1], *Petr Motlicek*[1], *Hervé Bourlard*[1,2]

[1]Idiap Research Institute, Martigny, Switzerland
[2]Ecole Polytechnique Fédérale de Lausanne, Switzerland

srikanth.madikeri, petr.motlicek, herve.bourlard@idiap.ch

Abstract

In this paper, we develop Automatic Speech Recognition (ASR) systems for multi-genre speech recognition of low-resource languages where training data is predominantly conversational speech but test data can be in one of the following genres: news broadcast, topical broadcast and conversational speech. ASR for low-resource languages is often developed by adapting a pre-trained model to a target language. When training data is predominantly from one genre and limited, the system's performance for other genres suffer. To handle such out-of-domain scenarios, we employ multitask adaptation by using auxiliary conversational speech data from other languages in addition to the target-language data. We aim to (1) improve adaptation through implicit data augmentation by adding other languages as auxiliary tasks, and (2) prevent the acoustic model from overfitting to the dominant genre in the training set. Pre-trained parameters are obtained from a multilingual model trained with data from 18 languages using the Lattice-Free Maximum Mutual Information (LF-MMI) criterion. The adaptation is performed with the LF-MMI criterion. We present results on MATERIAL datasets for three languages: Kazakh and Farsi and Pashto.

Index Terms: Lattice Free MMI, low-resource speech recognition, multitask learning

1. Introduction

In the MATERIAL (Machine Translation for English Retrieval of Information in Any Language[1]) program, ASR systems for low-resource languages are trained on predominantly conversational speech, but tested on speech from multiple genres: conversational speech (CS), news broadcast (NB) and topical broadcast (TB). For such tasks, an ASR that generalizes better across multiple genres despite the constraints imposed on the training data is desirable. Owing to the low-resource nature of the target languages, a common approach is to adapt a pre-trained model to the target language [1, 2]. Multilingual modelling is a common technique used to boost training resources for the acoustic model [3, 4, 5, 6]. In the Babel program [7], multilingual models were trained using data from all languages in the program [8, 9], which proved to be effective on both seen and unseen languages in training.

In [10, 11], adaptation of pre-trained Lattice Free-Maximum Mutual Information Criterion (LF-MMI) models was shown to be effective for ASR on out-of-domain data. In [12], multilingual models trained with the LF-MMI were shown to outperform monolingual models on both Babel and Globalphone datasets. In this paper, we show the effectiveness of adapting such multilingual LF-MMI models to MATERIAL's multi-genre test condition. Compared to training monolingual models with LF-MMI, adaptation of multilingual LF-MMI models perform significantly better across all genres. Similar to [11], we adapt the multilingual model by adding new language-specific output layers, even in the case where languages were seen during multilingual training.

Given a target-language, ASR can be trained by simply adapting existing language specific layers in the model, or adding new layers to be trained during adaptation. In the latter case, typically the learning rate on the pre-trained layers is a fraction of the learning on newly added layers (e.g. one tenth of the learning rate of the new layers [10]). Since the amount of adaptation data is limited (few tens of hours of speech) and mostly from a single domain (CS), the model tends to adapt well towards the genre predominant in the training data.

Unlike CS, broadcast speech data for many languages are available in the open source domain. Thus, to improve the performance on broadcast data one can further perform semi-supervised training (SST) [13, 14, 15, 16, 17]. Moreover, as shown in [16], improving the seed model can provide a considerable boost to the final performance on broadcast data with SST. Thus, we propose a simple approach using multitask learning that can provide a better starting point for techniques such as SST. The goal of applying multitask learning for adaptation (or transfer learning in the case where the target language is unseen during multilingual training) is to use auxiliary tasks as competing objectives to boost the adapted model's out-of-domain performance. Given the success of multilingual LF-MMI training, we extend it to target language adaptation as well. In this case, we consider models pre-trained with multilingual LF-MMI with 18 languages. The model is adapted, also with the LF-MMI criterion, along with other languages that are not necessarily our target. We refer to this technique as multitask adaptation (MTA), while the conventional adaptation of pre-trained models is referred to as Single Task Adaptation (STA). On MATERIAL datasets, we show that by replacing STA with MTA, one can achieve relative improvements in Word Error Rate (WER) of up to 7.1%. We will release the MTA adaptation code as part of the Babel multilingual recipe in Pkwrap [18] to adapt both Kaldi and Pytorch [19] acoustic models trained with LF-MMI [2].

The rest of the paper is organized as follows: in Sections 2 and 3, multilingual LF-MMI training and our multitask adaptation method are described, respectively. In Section 4, we detail the proposed approach of multi-task adaptation. In Section 5, experimental details and results are presented.

[1] https://www.iarpa.gov/index.php/research-programs/material

[2] https://github.com/idiap/pkwrap/tree/master/egs/multilang/babel/

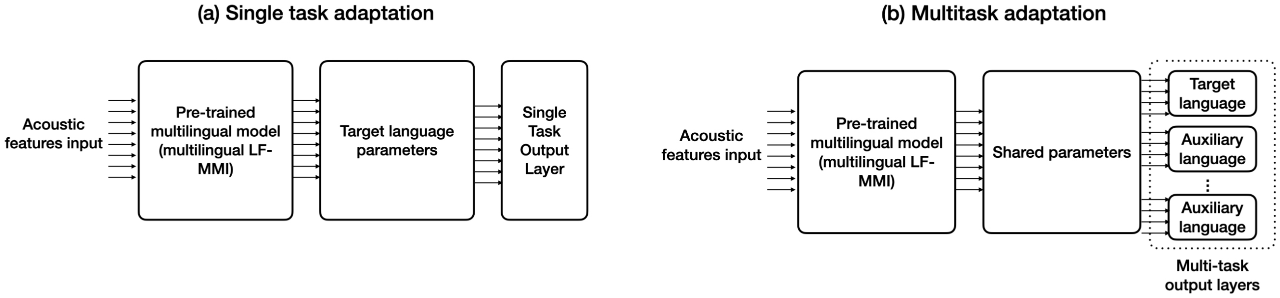


Figure 1: *(a) illustration of typical adaptation of pre-trained model to a target-language. (b) illustration of the proposed multitask adaptation with target language as one of the tasks. The target language shares parameters with auxiliary tasks (other languages used during adaptation).*

2. Multilingual LF-MMI

In [12], a multitask setup to train multilingual acoustic models with LF-MMI was introduced. The LF-MMI criterion provides state-of-the-art performance for hybrid ASR systems. LF-MMI provides a sequence discriminative training criterion, wherein each sequence (typically, an utterance of speech) is evaluated by two values: the numerator which computes the probability of the observation given the groundtruth, and the denominator which computes the probability over all possible sequences. The latter is computed with a graph, referred to as the denominator graph, trained from a phone Language Model (LM) [20]. The phone LM is trained from transcripts in the training data. In multilingual LF-MMI, the acoustic model shares parameters across languages, and there is one output layer for each language in the training dataset. Each language has its own denominator graph during training.

The performance of multilingual models on the Babel datasets is well established with standard Time Delay Neural Networks (TDNN) [21]. In this paper, we improve the model capacity of the AM by using the CNN-TDNN-F architecture (Convolutional Neural Networks and Factorized TDNNs) trained with 18 languages obtained from Babel and MATERIAL datasets (as opposed to only 14 in our previous work), thereby learning better representations suitable for cross-lingual learning [22, 23]. The list of datasets used for trained are given in Table 1. Note that we only refer to the multitask version of multilingual training in this paper, where each language in training has a separate output layer.

We apply transfer learning on this multilingual model to languages recently considered in the MATERIAL program: Pashto, Farsi and Kazakh. Out of the three, two languages, Pashto and Kazakh, overlap with the 18 languages used for multilingual training. Farsi is treated as an unseen language. The adaptation is carried out in a fashion similar to [10]. We do not freeze all the layers in the multilingual model, but fix a learning rate factor on the pre-trained layers. To adapt to each language, a learning rate factor of 0.1 was used. In addition to the pre-trained layers we also add additional target language-specific layers. To control the number of model parameters, we use TDNN-F layers [24]. The LF-MMI criterion is used for adaptation.

3. Multitask adaptation

In this section, we describe the proposed multitask approach. To motivate our approach we provide the following reasoning: in order to improve the AM for low-resource languages, multilingual modelling is often considered useful. Similarly, when adapting a well-trained acoustic model to a target language, one can employ a similar strategy by adapting multiple languages at the same time despite our interest being in only one of the languages. As mentioned earlier, we refer to this type of adaptation as Multitask adaptation (MTA). To contrast with MTA, we will refer to the conventional adaptation of pre-trained models to a target language as Single Task Adaptation (STA). In Natural Language Processing tasks, where using pre-trained models is quite common, MTA of pre-trained models has been shown to be effective [25]. Figure 1 illustrates the difference between STA and MTA.

Multitask learning [26, 27] has several well-documented advantages. Two important advantages that we consider here are implicit data augmentation and ability to reduce the risk of overfitting. When adapting pre-trained models to low-resource languages, we observed that despite heavy regularization through high dropout rates, the model performance saturates. To avoid such saturation we use the regularizing effect of adding new languages. Multitask learning for regularization has already been applied in different contexts. In LF-MMI training, it is common to use cross-entropy objective function as an auxiliary objective. In end-to-end ASR training, using multiple objective functions has been shown to be useful [28].

In addition, the presence of more data from different languages is well-known to improve speech models [29, 30, 31]. Thus, we hypothesize that adapting a pre-trained model to multiple languages instead of just the target language can be more beneficial to the performance on out-of-domain data. In this work, we consider four languages for MTA: Kazakh, Farsi, Pashto and Turkish. The first three are target languages, and Turkish is included due to its linguistic proximity to Kazakh (among the Babel datasets used in this work). In order to balance the trade-off between the adaptation speed and multi-task adaptation benefits, we do not consider more than four languages.

4. Experiments

We first evaluate the performance of the improved multilingual model on four languages from Babel: Tagalog (TGL), Swahili (SWA), Zulu (ZUL) and Turkish (TUR). The evaluation setup for Babel is the same as [12]. Then, we report the results on three languages in the MATERIAL program: Farsi, Kazakh and Pashto.

Assamese	Bengali	Cantonese	Haitian
Kazhak	Kurmanji Kurdish	Lao	Lithuanian
Pashto	Somali*	Swahili	Tagalog
Tamil	Telugu	Tok Pisin	Turkish
Vietnamese	Zulu		

Table 1: *Babel [7] and MATERIAL (marked with *) datasets used for multilingual training. The language names are sorted in alphabetical order.*

Layer	Parameter
CNN-1	64 filters
CNN-2	64 filters
CNN-3	128 filters + height subsampling
CNN-4	128 filters
CNN-5	256 filters + height subsampling
CNN-6	256 filters
TDNN-F	1536 dim, 256 dim BN
TDNN-F x 7	1563 dim + 0.66 bypass scale
Bottleneck layer	512 dimension

Table 2: *Description of the architecture of the multilingual CNN-TDNN-F model. The architecture is a modification of a similar model found in standard Kaldi recipes (`egs/librispeech/s5/local/chain/tuning/run_cnn_tdnn_1a.sh`). (dim: dimension, BN: bottleneck)*

4.1. Model training

The multilingual model was trained with the 18 languages given in Table 1. For all Babel datasets, only conversational speech data was used for training. We trained a 14-layer CNN-TDNN-F (Convolutional Neural Network followed by Factorized Time-delay Neural Networks [24]). The model architecture is given in Table 2. We used hybrid LF-MMI to train the model, with a weight of 1/18 for each language. The model takes as input 40 dimensional MFCC features and online i-vectors ([32, 33]). Three-fold speed-perturbation was applied to the training data.

To generate alignments for training, a HMM/GMM system was trained with PLP+pitch (a concatenation of Perceptual Linear Prediction and pitch) features using the standard recipe for Babel datasets in Kaldi [34]. The lexicon provided with the dataset was used. The alignments generated were used to create supervision lattices for LF-MMI training. The acoustic model was trained for 6 epochs with an exponentially decaying learning rate schedule with an initial learning rate of 0.001 and final learning rate of 0.0001. A dropout schedule with the following parameters was used: from 20% to 50% of the iterations, the dropout was increased from 0.0 to 0.25, and then was gradually decreased to 0.0 for the rest of the iterations. A continuous version of a dropout was used [34]. We used Kaldi for all our experiments.

4.2. Performance on Babel

The performance of the multilingual model on four languages is presented in Table 3. WERs are reported on dev10h test set. We also refer to performance reported in [35] to compare with our baseline monolingual systems. As reported in [12], the multilingual model trained with 14 languages is significantly better than the monolingual LF-MMI system. Relative improvements of up to 13.6% (SWA) was achieved. From the results with the CNN-TDNN-F model, it is clear that the multilingual training can further benefit with increased model capacity. The CNN-TDNN-F model improves further by 6.6% for TGL, 7.1% for SWA.

System	TGL	SWA	TUR	ZUL
Monolingual TDNN [12]	45.3	38.7	47.2	53.5
BLSTM [35]	46.3	38.3	-	61.1
Multilingual models				
TDNN (14 languages) [12]	42.2	33.6	43.9	50.8
CNN-TDNN-F (18 languages)	39.4	31.2	40.8	48.5

Table 3: *Comparison of performance of multilingual LF-MMI models on four languages in the Babel dataset. Word Error Rates (WER) on dev10h are reported. We also compare our results with [35] as reference to other multilingual models with similar datasets.*

Parameter	Pashto	Kazakh	Farsi
Training data (h)	78.4	49.8	36.3
Test data (CS, NB, TB) (h)	16.4	11.2	9.5
Vocabulary	239k	580k	1.7M
LM (words)	816k	184M	1.3B
LM Perplexity (3-gram)	560	789	786

Table 4: *Statistics of the MATERIAL test sets for Pashto, Kazakh and Farsi. Train and test data duration are computed after segmentation. The segmentation is taken from groundtruth. LM perplexities are calculated with the LM trained on all text available for the language and evaluated on only broadcast data transcripts.*

4.3. MATERIAL datasets

We consider three MATERIAL datasets: Kazakh, Pashto and Farsi. The first two languages are also part of the Babel datasets used for multilingual training while Farsi is an unseen language.

Language model for each dataset is trained as follows: for each language text obtained from web-crawl is available for language model. The web-crawl text is cleaned (punctuation and out-of-language word) and a 3-gram model is trained with SRILM [36] along with the training transcripts. We use Kneser-Ney smoothing with parameters 0, 1 and 2. This consistently gave us the best trade-off between language model perplexity and size. This language model is used for decoding NB and TB audio. For CS, we interpolate the LM with a 3-gram LM trained only with training transcripts. An interpolation weight of 0.9 on the latter is used [37]. The vocabulary for each language is chosen based on the web crawl text and training transcripts. While all words in the training transcripts are included, only words that appear at least 5 times in the web crawl are chosen as a part of the vocabulary. Graphemic lexicon was used for all the

	System	*Seen* languages						*Unseen* language		
		Pashto			Kazakh			Farsi		
		CS	NB	TB	CS	NB	TB	CS	NB	TB
(a)	Monolingual TDNN-F	47.2	47.0	54.8	44.3	29.4	36.2	50.7	56.6	49.7
(b)	Monolingual CNN-TDNNF	46.9	44.2	51.3	39.7	25.9	30.9	43.2	42.4	48.9
(c)	STA	41.9	43.6	48.1	39.2	23.4	26.6	37.0	36.6	41.1
(d)	MTA	41.8	**40.5**	**45.4**	38.9	**21.9**	**25.4**	36.9	**35.3**	**40.1**
(e)	Fusion (c+d)	40.8	40.7	45.2	37.6	21.6	24.7	35.3	33.8	38.6

Table 5: *Comparison of performance of adaptation with multilingual LF-MMI models to three MATERIAL datasets. Word Error Rates (WER) are reported. CS: Conversational speech, NB: News Broadcast, TB: Topical Broadcast, STA: Single task adaptation, MTA: Multitask adaptation*

three languages. All words in Kazakh were lower-cased. The statistics of training data is given in Table 4.

Two experiments were performed on the MATERIAL languages: (1) STA (adaptation of the multilingual CNN-TDNN-F model to the target language), and (2) MTA (multitask adaptation of the same pre-trained model to several target languages, simultaneously). We also used Babel Turkish as an additional language for MTA. The adaptation was carried out by setting a learning rate factor of 0.1 on the pre-trained layers. Additional 9 layers of TDNN-F was added to adapt to each target language. All but the first TDNN-F component had a context of 3. The first TDNN-F layer takes as input the output of the bottleneck layer of the multilingual model. The same network architecture was used for both STA and MTA. Each output layer in MTA had a learning rate factor of 0.25 (i.e. all languages were weighted equally). An exponentially decaying learning rate schedule was used with initial learning rate of 0.001 and final learning rate of 0.0005. A different dropout schedule was used during adaptation: dropout rate was kept to 0.0 for the first 5% of the iterations, then increased to 0.25 until 60% of the iterations, followed by reduction to 0.0 until the final iteration.

4.4. Performance on MATERIAL datasets

The results are presented in Table 5. First we compare the results of monolingual systems with systems adapted from the multilingual model. Considerable improvements are observed for all 3 languages. The benefits of adapting a multilingual model with STA is shown by relative improvements obtained up to 15.9% (Farsi, TB) compared to the best monolingual system. All systems performed the worst on the TB compared to other genres owing to the difficulty of the genre (mostly in terms of acoustic conditions and vocabulary). Adapting any of the three target languages provides significant performance boost for all genres.

With MTA, improvements in the broadcast genre (i.e. NB and TB) were observed for all languages. The results demonstrate that MTA can be beneficial compared to STA for out-of-domain data. Note that for both STA and MTA the same model configuration is used. Relative improvements ranging from 2.5% (TB in Farsi) to 7.1% (NB in Pashto) are observed for the broadcast genre. For in-domain data (CS), we only observed marginal gain in performance. However, to verify if the acoustic model trained with MTA is different to that obtained with STA, we performed a simple system fusion experiment. Improvements observed on 8 out of the 9 subsets suggest that MTA learns representations different to that learnt with STA. Even though the difference between STA and MTA performances are negligible for the CS genre, the fusion of the two systems provided relative improvements between 2.4% (Pashto) and 4.4% (Farsi). For NB in Pashto, there is a slight degradation in performance (from 40.5% to 40.7%) suggesting that the acoustic representation obtained with MTA can sometimes be considerably better for broadcast data than that obtained with STA.

5. Summary

We presented results on four Babel languages with multilingual LF-MMI training. We showed that multilingual LF-MMI scales well with increased model capacity, and with the number of languages used during training. We demonstrated the usefulness of such pre-trained models for multi-genre speech recognition on the MATERIAL dataset for three languages: Pashto, Kazakh and Farsi. Consistent improvements were obtained for both seen and unseen languages. To further improve the performance on broadcast data we proposed multitask adaptation. Relative improvements ranging between 2.5% and 7.1% were obtained compared to the conventional adaptation on news and topical broadcast.

6. Acknowledgements

The research is based upon the work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via AFRL Contract FA8650-17-C-9116. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

We thank Joel Barry (ISI, USC) for sharing the web-crawl text for the MATERIAL languages used in this paper.

7. References

[1] F. Grézl, M. Karafiát, and K. Veselỳ, “Adaptation of multilingual stacked bottle-neck neural network structure for new language,” in *Proc. of ICASSP 2014*, pp. 7654–7658.

[2] D. Bagchi and W. Hartmann, "Learning from the best: A teacher-student multilingual framework for low-resource languages," in *Proc. of ICASSP 2019*, pp. 6051–6055.

[3] D. Imseng, J. Dines, P. Motlicek, P. N. Garner, and H. Bourlard, "Comparing different acoustic modeling techniques for multilingual boosting," in *In Proc. of Interspeech*, 2012, pp. 1191–1194.

[4] P. Motlicek, D. Imseng, B. Potard, P. N. Garner, and I. Himawan, "Exploiting foreign resources for DNN-based ASR," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2015, no. 1, pp. 1–10, 2015.

[5] P. Motlicek, F. Valente, and P. N. Garner, "English spoken term detection in multilingual recordings," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[6] D. Imseng, P. Motlicek, P. N. Garner, and H. Bourlard, "Impact of deep MLP architecture on different acoustic modeling techniques for under-resourced speech recognition," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013, pp. 332–337.

[7] M. Harper, "The BABEL program and low resource speech technology," *In Proc. of Automatic Speech Recognition and Understanding*, 2013.

[8] M. Karafiát *et al.*, "Multilingual BLSTM and speaker-specific vector adaptation in 2016 BUT Babel system," in *IEEE Spoken Language Technology Workshop (SLT)*, 2016, pp. 637–643.

[9] K. M. Knill *et al.*, "Investigation of multilingual deep neural networks for spoken term detection," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013, pp. 138–143.

[10] P. Ghahremani, V. Manohar, H. Hadian, D. Povey, and S. Khudanpur, "Investigation of transfer learning for asr using lf-mmi trained neural networks," in *IEEE Automatic Speech Recognition and Understanding Workshop*, 2017, pp. 279–286.

[11] A. Vyas, S. Madikeri, and H. Bourlard, "Lattice-free MMI adaptation of self-supervised pretrained acoustic models," *In Proc. of ICASSP 2021*, pp. 6219–6223.

[12] S. Madikeri *et al.*, "Lattice-free maximum mutual information training of multilingual speech recognition systems," in *Proc. of Interspeech*, 2020, pp. 4746–4750.

[13] J. Ma, S. Matsoukas, O. Kimball, and R. Schwartz, "Unsupervised training on large amounts of broadcast news data," in *Proc. of ICASSP 2006*, pp. 1056–1059.

[14] K. Veselỳ, M. Hannemann, and L. Burget, "Semi-supervised training of deep neural networks," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013, pp. 267–272.

[15] D. Imseng *et al.*, "Exploiting un-transcribed foreign data for speech recognition in well-resourced languages," in *Proc. of ICASSP*, 2014, pp. 2322–2326.

[16] B. Khonglah *et al.*, "Incremental semi-supervised learning for multi-genre speech recognition," in *Proc. of ICASSP*, 2020, pp. 7419–7423.

[17] V. Manohar, H. Hadian, D. Povey, and S. Khudanpur, "Semi-supervised training of acoustic models using lattice-free MMI," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4844–4848.

[18] S. Madikeri, S. Tong, J. Zuluaga-Gomez, A. Vyas, P. Motlicek, and H. Bourlard, "Pkwrap: a pytorch package for LF-MMI training of acoustic models," *arXiv preprint arXiv:2010.03466*, 2020.

[19] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *arXiv preprint arXiv:1912.01703*, 2019.

[20] D. Povey *et al.*, "Purely sequence-trained neural networks for asr based on Lattice-Free MMI," in *Proc. of Interspeech*, 2016, pp. 2751–2755.

[21] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. of Interspeech*, 2015, pp. 3214–3218.

[22] I. Medennikov, Y. Y. Khokhlov, A. Romanenko, I. Sorokin, A. Mitrofanov, V. Bataev, A. Andrusenko, T. Prisyach, M. Korenevskaya, O. Petrov *et al.*, "The STC ASR system for the voices from a distance challenge 2019." in *In Proc. of INTERSPEECH*, 2019, pp. 2453–2457.

[23] M. Karafiát, M. K. Baskar, I. Szöke, H. K. Vydana, K. Veselỳ, J. Černockỳ *et al.*, "BUT Opensat 2019 speech recognition system," *arXiv preprint arXiv:2001.11360*, 2020.

[24] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks." in *Proc. of Interspeech*, 2018, pp. 3743–3747.

[25] X. Liu, P. He, W. Chen, and J. Gao, "Multi-task deep neural networks for natural language understanding," *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4487–4496, 2019.

[26] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[27] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[28] J. Li, Y. Wu, Y. Gaur, C. Wang, R. Zhao, and S. Liu, "On the comparison of popular end-to-end models for large scale speech recognition," *Proc. of Interspeech*, pp. 1–5, 2020.

[29] N. T. Vu and T. Schultz, "Multilingual multilayer perceptron for rapid language adaptation between and across language families." in *In Proc. of Interspeech*, 2013, pp. 515–519.

[30] N. T. Vu *et al.*, "Multilingual deep neural network based acoustic modeling for rapid language adaptation," in *Proc. of ICASSP*, 2014, pp. 7639–7643.

[31] D. Imseng, P. Motlicek, H. Bourlard, and P. N. Garner, "Using out-of-language data to improve an under-resourced speech recognizer," *Speech communication*, vol. 56, pp. 142–151, 2014.

[32] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013, pp. 55–59.

[33] A. Senior and I. Lopez-Moreno, "Improving DNN speaker independence with i-vector inputs," in *Proc. of ICASSP*, 2014, pp. 225–229.

[34] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," IEEE Signal Processing Society, Tech. Rep., 2011.

[35] H. Inaguma, J. Cho, M. K. Baskar, T. Kawahara, and S. Watanabe, "Transfer learning of language-independent end-to-end ASR with language model fusion," in *Proc. of ICASSP*, 2019, pp. 6096–6100.

[36] A. Stolcke, "SRILM-an extensible language modeling toolkit," in *Seventh international conference on spoken language processing*, 2002.

[37] E. Boschee *et al.*, "Saral: A low-resource cross-lingual domain-focused information retrieval system for effective rapid document triage," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2019, pp. 19–24.

5.5 Paper 5: [Sri+17]

Semi-supervised Learning with Semantic Knowledge Extraction for Improved Speech Recognition in Air Traffic Control

Ajay Srinivasamurthy[1], *Petr Motlicek*[1], *Ivan Himawan*[1], *György Szaszák*[2], *Youssef Oualil*[2], *Hartmut Helmke*[3]

[1]Idiap Research Institute, Martigny, Switzerland
[2]Spoken Language Systems Group, Saarland University (UdS), Saarbrücken, Germany
[3]Institute for Flight Guidance, German Aerospace Center (DLR), Braunschweig, Germany

ajay.srinivasamurthy@idiap.ch, petr.motlicek@idiap.ch, ivan.himawan@idiap.ch, gszaszak@lsv.uni-saarland.de, youalil@lsv.uni-saarland.de, hartmut.helmke@dlr.de

Abstract

Automatic Speech Recognition (ASR) can introduce higher levels of automation into Air Traffic Control (ATC), where spoken language is still the predominant form of communication. While ATC uses standard phraseology and a limited vocabulary, we need to adapt the speech recognition systems to local acoustic conditions and vocabularies at each airport to reach optimal performance. Due to continuous operation of ATC systems, a large and increasing amount of untranscribed speech data is available, allowing for semi-supervised learning methods to build and adapt ASR models. In this paper, we first identify the challenges in building ASR systems for specific ATC areas and propose to utilize out-of-domain data to build baseline ASR models. Then we explore different methods of data selection for adapting baseline models by exploiting the continuously increasing untranscribed data. We develop a basic approach capable of exploiting semantic representations of ATC commands. We achieve relative improvement in both word error rate (23.5%) and concept error rates (7%) when adapting ASR models to different ATC conditions in a semi-supervised manner.

Index Terms: Speech Recognition, Air Traffic Control, Semi-supervised learning

1. Introduction

Air Traffic Control (ATC) involves spoken language communication between aircraft pilots and air traffic controllers, who guide aircraft to navigate safely in air and at airports. The intensive use of spoken language in ATC is natural and hence preferred in some ways, but it also hampers the introduction of higher levels of automation. Introduction of ASR (Automatic Speech Recognition) into ATC systems is an enabler for different levels of automation, reducing the efforts of air traffic controllers leading to significant gains in terms of reduced human effort and saved flight times.

Recently, Assistant based Speech Recognition (ABSR)[1] [1] that combines ASR with a radar based assistant system has been shown to be useful. An ABSR generates context information to reduce the search space for the speech recognizer and can reduce controller's workload by a factor of three [2], in addition to significant fuel savings [3] resulting from shorter flight times and increased operational efficiency. However, extending ABSR to real world operational environments is challenging for many reasons. To build robust ASR systems for each operating ATC environment, transcribed speech data is necessary, obtaining which is time and resource consuming. Owing to its global nature, ATC uses standardized English vocabulary and phraseology for communication. However, local variations in each ATC area exist due to local runways, waypoints, airlines, acoustic conditions, local English accents and the occasional use of local language words. Further, some of the local conditions (airlines, runways, waypoints) can also change over time and hence the ASR systems need regular maintenance. Due to continuous operation of ATC systems, an increasing amount of (untranscribed) speech and radar data is generated and is archived for flight safety reasons. The MALORCA[2] project has been constituted to address these issues and automate re-learning, adaptation and customization of ASR systems to new ATC environments. The main goal is to continuously update the ASR models in an unsupervised/semi-supervised manner by utilizing increasing amounts of speech data, while exploiting local acoustic, language and semantic constraints. In addition, data from other modalities such as radar can be used, which provide a context for the commands issued by the controllers to pilots.

ASR systems built to a specific domain ensure the best performance. However, in their absence, adapting out-of-domain (OOD) ASR models to a specific domain has been explored [4, 5]. In aviation, ASR is a known technology used with considerable success in training simulators [6]. Applying ASR to ATC domain has been previously explored [7], but the use of untranscribed data is a new challange. Semi-supervised learning methods [8] can be used to utilize the untranscribed data to improve and build domain specific ASR systems. A "first iteration" ASR built with limited training data can be used to automatically transcribe raw audio data, thus generating approximate transcriptions that can be used as additional training data. Data selection for semi-supervised learning [9] from such automatic transcripts then becomes a central task, where different confidence measures at frame, word and sentence level have been used and several methods have been proposed [10, 11, 12]. Semantics based confidence measures have received some attention in specific tasks related to spoken dialogue systems [13, 14]. However, the variation in semantics across different application domains of ASR motivates the need for domain specific semantic confidence measures.

In this paper, we explore the tasks associated with automatic deployment and adaptation of ASR models to a new ATC environment. We use a limited amount of transcribed data available from Vienna ATC area while also utilizing additional

[1] AcListant®: http://www.aclistant.de

[2] MAchine Learning Of speech Recognition models for Controller Assistance: http://www.malorca-project.de/

OOD data. We propose data selection methods to choose suitable training data from untranscribed speech from Vienna ATC area and discuss directions for further improvement of semi-supervised learning methods. ATC communication has a limited vocabulary with strong semantic restrictions. The goal of such communication is to ensure that the necessary commands from controllers to pilots are conveyed through spoken language. The commands are hence primary, while the exact spoken text is of secondary importance. Any improvements to an ASR system in such an application should be geared towards improving the accuracy of command recognition. Hence measures and approaches that can work with command semantics in addition to the commonly used phone and word levels are preferred. We also explore such methods in this paper.

2. Semi-supervised Learning: Methods

In this paper, (1) we build base ASR models using limited in-domain data from Vienna ATC area and out-of-domain data, (2) the base ASR models are then supervised-adapted to Vienna ATC area. Subsequently, (3) the ASR models are used for further semi-supervised learning experiments. We start by first describing the datasets used in the experiments.

2.1. Datasets

The speech data used in this paper has been recorded from Vienna approach sector and feeder controller. A part of the speech data is transcribed, with text and command transcriptions. The availability of a partial set of transcriptions provides us the right opportunity to explore semi-supervised learning methods to utilize the complete untranscribed data. Vienna ATC continuously records speech data and hence can provide increasing amounts of (untranscribed) speech data. At the moment, this data is not publicly available. The speech content of the dataset is similar to other publicly available ATC domain datasets such as the LDC ATC dataset [15] and ATCOSIM dataset [16].

While additional data from Vienna approach is expected, presently the dataset has over 20 hours of speech data from 46 different controllers (speakers). All the data was recorded from operational ATC environments in the second half of 2016 at a sampling rate of 8kHz. The data has been segmented into short utterances containing only a few (upto 5) controller commands (most utterances have just one command). A command from a controller is repeated by the pilots (readback), but the pilot replies are not recorded and stored since they are not relevant. While all recordings have speaker labels, only a part of the dataset is annotated by professional air traffic controllers with text and command transcripts using an in-house annotation tool.

For training the base ASR models, we use about 5 hours of transcribed data, which we term as VDev1. The transcriptions include text transcriptions of the speech utterance, along with a transcription of the command that the speech utterance conveys to the pilots. For testing, we use about 2 hours of transcribed data with 6 speakers, termed as VTest dataset. About 9 hours of untranscribed data termed as VDev2 is used for semi-supervised learning of models. The three datasets are disjoint and do not share any speakers across them, as described in Table 1.

Since the amount of transcribed data available from Vienna approach is limited, we utilize other available transcribed resources to train the ASR system. We hypothesize that the use of standard English datasets is useful for seed training an acoustic model. We pool 150 hours of speech data from the publicly available LIBRISPEECH [17], ICSI [18], AMI [19] and TED-LIUM [20] datasets, which have been extensively used for recognition of conversational speech. The speech data and accompanying transcripts (called MEGA) are used in conjunction with training data from Vienna approach.

Dataset	Source	Dur. (hr)	Speakers
VDev1	Vienna approach	5.1	13
VDev2	Vienna approach	9.1	24
VTest	Vienna approach	1.9	6
MEGA	LIBRISPEECH, AMI, ICSI, TED-LIUM	150	1043

Table 1: *Datasets, showing the source, duration and speakers*

2.2. Dictionary, Acoustic and Language models

We add all the possible in-domain words associated with Vienna ATC area (e.g. airlines and waypoints) to the standard CMU-Sphinx dictionary[3] to form an extended pronunciation dictionary for use with both acoustic and language models. There are hence no out-of-vocabulary words during training or testing.

DNN/HMM (Deep Neural Network Hidden Markov Model) acoustic models are the state of the art in speech recognition acoustic modeling. As reliable training of DNNs require significant amount of labeled data, we add the 150 hour MEGA dataset to the limited Vienna VDev1 dataset. We use the combined data to train a Gaussian Mixture Model based GMM/HMM acoustic model (AM). Using the state level alignments of the combined data using the GMM/HMM model, we train a DNN/HMM acoustic model (called the DNN-BASE).

The DNN-BASE acoustic model is then adapted to Vienna ATC domain using the VDev1 dataset. To adapt, we start from the DNN-BASE model, and first reinitialize and randomize the weights of the last layer of the DNN. The architecture and weights of the other layers are unchanged. We then retrain the entire network using VDev1 training dataset to obtain supervised-adapted DNN (DNN-SA). This way of reinitializing the last layer and retraining the complete network was found to be effective for supervised adaptation using in-domain data.

For decoding a test utterance, we use a trigram language model (LM) built using the transcripts of VDev1 (vocabulary size $\approx$ 700 words) to ensure that an in-domain Vienna specific language model is used. Together with the language model, the ASR system using DNN-BASE and DNN-SA with the trigram language model from VDev1 is called ASR-BASE and ASR-SA, respectively.

The standard vocabulary and phraseology used in ATC is an argument to construct a rule based Context-Free Grammar (CFG) that can be used to build a Vienna specific language model. However, in practice, the controllers often deviate from standards, and hence an N-gram statistical language model is used instead for recognition, while a CFG is used for concept extraction, as further described next.

2.3. Concept and Command extraction

The output from an ASR system is a sequence of words as spoken by the controller. We however then need to extract the controller command that the sequence contains. From the controller utterances we extract concepts and commands. Concepts include all meaningful words or expressions which are related to the controller command and the required action of the aircraft. Concepts basically include (i) the callsign composed of an airline identifier (International Civil Aviation Organization airline code) and a flight number, (ii) the command word or ex-

[3] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

pression itself, and (iii) the command attributes (usually target values for some flight parameters). This sequence of concepts forms a command. For example, the following utterance "*hello lufthansa eight echo kilo, start reduce your speed to two two zero knots*" contains the following concepts:

- DLH8EK (*lufthansa eight echo kilo* - callsign)
- REDUCE (*reduce* - command word)
- 220 (*two two zero* - speed attribute)

The command in its semantic form is hence: DLH8EK REDUCE 220. In order to extract the concepts from the utterance, we use a CFG that models controller phraseology. Normally phraseology is highly standardized, i.e. the controllers are fairly bound on how they formulate a command. All possible command words or expressions have an entry in the CFG, for modelling standard phraseology and often used deviation forms. Each semantic slot for the command is tagged in the CFG, and hence, transducing [21] a transcript hypothesis from the ASR over the CFG results in an XML tagged version as follows (using the previous example): *hello* <callsign> <airline> *lufthansa* </airline> <flightnumber> *eight echo kilo* </flightnumber> </callsign> *start* <commands> <command="reduce"> *reduce your speed to* <speed> *two two zero* </speed> *knots* </command> </commands>. If transductions fail (due to a deviation in phraseology not modelled by the CFG or due to ASR errors), the command extractor returns "NO_CALLSIGN" if the callsign is missed, and "NO_CONCEPT", if the command word or the command attribute could not be recovered.

Thus, given a speech utterance by a controller, we obtain a plain text hypothesis (sequence of words as they were spoken), an XML tagged version of hypothesis (tagged with semantic concepts), and the command hypothesis.

2.4. Semi-supervised learning

Semi-supervised learning aims to exploit the untranscribed data available in VDev2 dataset to improve the ASR models. Starting with the supervised-adapted ASR-SA system, the approach we use in this paper consists of three steps: transcript generation, data selection, and semi-supervised training.

2.4.1. Transcript generation

First we use the system adapted to VDev1 (ASR-SA) to generate the text and command transcripts for the data in VDev2. These automatically generated transcripts are used for further experiments.

2.4.2. Data selection

The automatically generated transcripts along with speech in VDev2 can be used as training data. However, these transcripts might have errors and those should be excluded from training, which is a problem often termed as data selection. Data selection is done by assigning confidence scores to ASR outputs, so that high confidence transcripts (and corresponding utterances) can be selected for further experiments. We explore two different data selection strategies, one that uses word level confidences and another that uses concept and command level confidences. Both data selection methods aim to utilize automatically transcribed data to provide additional training resources.

Word confidence: A logistic regression model is built with word-lattice derived features using the VDev1 transcribed data. The features include the posterior probability of a word obtained from Minimum Bayes Risk (MBR) decoding [22], word length, competing words, and frames per character ratio. The trained logistic regression model is applied with the same features extracted from the decoding word-lattices of VDev2 and output confidences (ranging from 0 to 1) per word are obtained. Utterance level confidence is then obtained as the average word-confidence of the words in the output. The utterance-confidence values are sorted and a threshold is used to select high confidence data into a subset VDev2-W of the automatically transcribed VDev2 dataset.

System	Training dataset	#Sen.	WER (%)	CER (%)
ASR-DEV1	VDev1	2143	12.3	38.6
ASR-BASE	MEGA + VDev1	3861	13.3	41.4

Table 2: *Baseline results on evaluation with VTest dataset and using an LM built with VDev1 transcripts, showing the number of senones (#Sen.), Word (WER) and Concept (CER) error rate.*

Concept confidence: Since the output commands are more relevant than the plain text ASR hypotheses, a data selection method that can incorporate a confidence measure based on output command hypothesis is preferred. We hypothesize that an accurate ASR output would result in an accurate command hypothesis generated by the command extractor. In case the command extractor is unable to decipher a valid command from the ASR output, it implies an erroneous automatic transcription. We base our data selection method on this premise, and exclude all automatic transcriptions that contain NO_CALLSIGN or NO_CONCEPT (and hence indicate the failure of the command extractor to extract a meaningful and valid concept and command hypothesis) as output command, to obtain a subset VDev2-C. Note that a valid output from the command extractor does not always imply an accurate command hypothesis. Nevertheless, we observed that command recognition is mostly accurate when the command extractor does not return NO_CONCEPT/NO_CALLSIGN. Without ground truth command transcripts, we explore this method as a first step towards command semantics based data selection.

2.4.3. Semi-supervised training

With either data selection methods, we combine VDev1 with the selected subset of VDev2 (either VDev2-W or VDev2-C) and their automatically obtained transcripts to form a larger adaptation dataset. Based on our previously published ideas, we explore adapting either the AM [23, 24], LM [25], or both using this adaptation dataset. To adapt only the AM, similar to training the DNN-SA, we reinitialize the last layer of DNN-BASE model and retrain the complete network with the adaptation dataset, while using the LM built with only VDev1. To adapt only the LM, we use the supervised-adapted DNN-SA acoustic model with a 3-gram LM built with the adaptation dataset. To adapt both AM and LM, we adapt DNN-BASE with the combined dataset and use a 3-gram LM built with the adaptation dataset. The ASR systems with semi-supervised adaptation using word and concept based confidences are termed ASR-SSA-W and ASR-SSA-C, respectively.

2.5. Evaluation measures

The most relevant metric of performance for ATC applications is at the command semantics level. However, since the ASR system outputs hypothesis at both word level and command level, we report the commonly used Word Error Rate (WER) and the Concept Error Rate (CER). For the CER, we discard all the semantically irrelevant words with respect to the command type from the output text hypothesis and match only the con-

System	Selection method	Adaptation dataset (Duration)	WER (%)			CER (%)		
			AM	LM	AM+LM	AM	LM	AM+LM
ASR-SA	—	VDev1 (5.1 hr)	10.0	—	—	37.5	—	—
ASR-SSA-none	None	+ VDev2 (9.1 hr)	9.6	9.8	9.6	36.6	37.3	36.9
ASR-SSA-W	Word	+ VDev2-W (7.2 hr)	9.6	9.8	**9.4**	36.8	36.7	37.0
ASR-SSA-C	Concept	+ VDev2-C (7 hr)	9.8	9.8	9.5	37.1	36.1	**35.9**

Table 3: *Results on evaluation with VTest dataset for supervised (ASR-SA) and semi-supervised methods (ASR-SSA-none, ASR-SSA-W, ASR-SSA-C), showing the selection method, adaptation dataset used, AM, LM or AM+LM adaptation, and the measures Word (WER) and Concept (CER) error rate. All acoustic models have 3861 senones at the output. Since the default LM is built with VDev1, LM adaptation is not applicable to ASR-SA. The best WER and CER is marked in bold.*

cepts, by treating the command word and its attribute together. For example, supposing a ground truth transcript of DLH8EK REDUCE_230 and a hypothesis of DLH8EK REDUCE_220, the CER is 50%, since the callsign is correctly hypothesized while command attribute is wrong. Owing to its inclusion of semantics, CER is a stricter measure than the WER.

3. Experiments

The speech recognition experiments are done using the Kaldi speech recognition toolkit [26].

3.1. Experimental setup

The GMM/HMM acoustic model is trained in a conventional fashion and consists of ≈3900 senones. In all the training cases, 50K Gaussians were added to the GMM/HMM model using diagonal covariance matrices. As input features, we applied 13 dim MFCCs accompanied with their delta and acceleration coefficients (39 dim feature vector), along with fMLLR transforms for speaker adaptive training. For the DNN/HMM model, the DNN comprises 4 layers: 351 dim input layer (9 stacked MFCC vectors with a context of 4 frames around the centered frame), hidden layers of 1200 nodes and output layer trained to discriminate among senones to estimate senone posterior probabilities. The DNN is trained to minimize frame-level cross entropy. To establish baselines, we additionally train smaller GMM/HMM (consisting of ≈2100 senones) and DNN/HMM acoustic models with VDev1 without utilizing the OOD data.

Starting from the DNN-BASE model, the supervised-adapted DNN-SA is trained as described in Section 2.2, with the same architecture. The semi-supervised methods follow the process described in Section 2.4.3, adapting either the AM, LM or both, using word confidence (ASR-SSA-W) or concept confidence (ASR-SSA-C). An average word confidence threshold of 0.95 is used for utterance selection, selecting (from VDev2) 7.2 hours of speech into VDev2-W. Command confidence based selection retains 7 hours of speech in VDev2-C data subset. In order to compare the performance of data selection, we also report results with no data selection (i.e. using all of VDev2), termed as ASR-SSA-none.

3.2. Results

We report results only with DNN/HMM acoustic models since they provided a better performance than the GMM/HMM counterparts. The baseline results are shown in Table 2 while the results of supervised and semi-supervised (AM, LM) adaptation are shown in Table 3. Both tables show the evaluation results on VTest dataset, with the baseline supervised training with only VDev1 (ASR-DEV1), VDev1 combined with MEGA (ASR-BASE), supervised adaptation of DNN-BASE with VDev1 (ASR-SA) and the two semi-supervised methods ASR-SSA-W (word confidence) and ASR-SSA-C (concept confidence), in addition to ASR-SSA-none (no data selection). While using only VDev1 to build smaller models seems to perform better in baselines in Table 2, the use of MEGA dataset helps building generalizable larger models that outperform with supervised adaptation as seen from ASR-SA (WER: 10.0%, which is an 18.7% relative decrease compared to 12.3% WER of ASR-DEV1).

Table 3 shows that the addition of automatically transcribed data for training is useful and improves performance over ASR-SA in all cases. It also shows the advantage of AM and LM adaptation, while adapting both AM and LM leads to better WER. The results also indicate that AM adaptation is marginally better than LM adaptation to improve WER, while such an observation does not extend to CER.

The ASR system built without data selection (ASR-SSA-none) shows a 4% relative improvement in WER over ASR-SA, while further data selection methods provide marginal improvement. The best performing WER of 9.4% (6% relative improvement over ASR-SA) is with AM+LM adaptation using word confidence based data selection (ASR-SSA-W), while the best performing CER of 35.9% (relative 4% improvement over ASR-SA, with 35 more concepts correctly hypothesized in total) is with AM+LM adaptation using concept confidences for data selection (ASR-SSA-C). This indicates that concept confidence measures help to achieve lower CER, while word confidence measures improve WER.

4. Conclusions

We built domain specific ASR models for controller pilot communication for Vienna approach by utilizing 150 hours of OOD data and adapting with 5 hours of in-domain transcribed data. We proposed data selection methods using word level and concept level confidences to benefit from cheaply available untranscribed in-domain data. This complemented transcribed in-domain data, enabling an adaptation of both acoustic and language models. Exploiting OOD data, plus complementing transcribed data with untranscribed in-domain data through data selection gives a relative reduction of WER by 23.5% (using word confidences) and CER by 7% (using concept confidences), when compared to using only in-domain transcribed data (ASR-DEV1, WER: 12.3%, CER: 38.6%). In the future, we will explore using additional amounts of untranscribed data for data selection. We also plan to integrate additional semantic information and other modalities such as radar data to develop improved training (such as transfer learning, sequence training with concept error metrics) and data selection methods for semi-supervised learning.

Acknowledgements: The work presented was conducted as a part of MALORCA project, funded by the SESAR Joint Undertaking (Grant Agreement No. 698824), under EU Horizon 2020 Research and Innovation programme. The authors thank Austro Control, Vienna for providing data from Vienna approach.

5. References

[1] H. Helmke, J. Rataj, T. Mühlhausen, O. Ohneiser, H. Ehr, M. Kleinert, Y. Oualil, M. Schulder, and D. Klakow, "Assistant-based speech recognition for ATM applications," in *Proc. of 11th USA/Europe Air Traffic Management Research and Development Seminar (ATM 2015)*, Jun. 2015.

[2] H. Helmke, O. Ohneiser, T. Mühlhausen, and M. Wies, "Reducing Controller Work-load with Automatic Speech Recognition," in *Proc. of the IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, Sacramento, USA, 2016.

[3] H. Helmke, O. Ohneiser, J. Buxbam, and C. Kern, "Increasing ATM Efficiency with Assistant Based Speech Recognition," in *Proc. of the 13th USA/Europe Air Traffic Management Research and Development Seminar*, Seattle, USA, 2017 (to appear).

[4] K. Yao, D. Yu, F. Seide, H. Su, L. Deng, and Y. Gong, "Adaptation of context-dependent deep neural networks for automatic speech recognition," in *Proc. of the IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2012, pp. 366–369.

[5] P. C. Woodland, X. Liu, Y. Qian, C. Zhang, M. J. Gales, P. Karanasou, P. Lanchantin, and L. Wang, "Cambridge university transcription systems for the multi-genre broadcast challenge," in *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015, pp. 639–646.

[6] D. Schäfer, "Context-sensitive speech recognition in the air traffic control simulation," Ph.D. dissertation, University of Armed Forces, Munich, 2001.

[7] H. D. Kopald, A. Chanen, S. Chen, E. C. Smith, and R. M. Tarakan, "Applying automatic speech recognition technology to air traffic management," in *Proc. of the IEEE/AIAA 32nd Digital Avionics Systems Conference (DASC)*, 2013.

[8] J. Glass, "Towards unsupervised speech processing," in *Proc. of the 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)*. IEEE, 2012, pp. 1–4.

[9] T. Drugman, J. Pylkkönen, and R. Kneser, "Active and Semi-Supervised Learning in ASR: Benefits on the Acoustic and Language Models," in *Proc. of Interspeech*, 2016, pp. 2318–2322.

[10] D. Imseng, B. Potard, P. Motlicek, A. Nanchen, and H. Bourlard, "Exploiting un-transcribed foreign data for speech recognition in well-resourced languages," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 2322–2326.

[11] R. Zhang and A. I. Rudnicky, "A new data selection approach for semi-supervised acoustic modeling," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2006.

[12] S. Li, Y. Akita, and T. Kawahara, "Semi-supervised Acoustic Model Training by Discriminative Data Selection from Multiple ASR Systems' Hypotheses," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 24, no. 9, pp. 1520–1530, Sep. 2016.

[13] R. Sarikaya, Y. Gao, M. Picheny, and H. Erdogan, "Semantic confidence measurement for spoken dialog systems," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 4, pp. 534–545, 2005.

[14] S. S. Pradhan and W. H. Ward, "Estimating semantic confidence for spoken dialogue systems," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2002.

[15] J. Godfrey, "Air Traffic Control Complete LDC94S14A," DVD, 1994. [Online]. Available: http://catalog.ldc.upenn.edu/LDC94S14A

[16] K. Hofbauer, S. Petrik, and H. Hering, "The ATCOSIM Corpus of Non-Prompted Clean Air Traffic Control Speech," in *Proc. of the 6th International Conference on Language Resources and Evaluation (LREC)*, 2008.

[17] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[18] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters, "The ICSI meeting corpus," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2003.

[19] J. Carletta, "Announcing the AMI meeting corpus," *The ELRA Newsletter*, vol. 11, no. 1, pp. 3–5, 2006.

[20] A. Rousseau, P. Deléglise, and Y. Estève, "Enhancing the TED-LIUM Corpus with Selected Data for Language Modeling and More TED Talks," in *Proc. of the 9th International Conference on Language Resources and Evaluation (LREC)*, 2014, pp. 3935–3939.

[21] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri, "OpenFst: A general and efficient weighted finite-state transducer library," in *Proc. of the International Conference on Implementation and Application of Automata*. Springer, 2007, pp. 11–23.

[22] H. Xu, D. Povey, L. Mangu, and J. Zhu, "An improved consensus-like method for Minimum Bayes Risk decoding and lattice combination," in *Proc. of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2010, pp. 4938–4941.

[23] P. Motlicek, "Automatic Out-of-Language Detection Based on Confidence Measures Derived from LVCSR Word and Phone Lattices," in *Proc. of the 10th Annual Conference of the International Speech Communication Association*, 2009.

[24] D. Imseng, B. Potard, P. Motlicek, A. Nanchen, and H. Bourlard, "Exploiting un-transcribed foreign data for speech recognition in well-resourced languages," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2014, pp. 2322 – 2326.

[25] G. Lecovre, J. Dines, T. Hain, and P. Motlicek, "Supervised and unsupervised web-based language model domain adaptation," in *Proc. of Interspeech*, 2012, pp. 131–134.

[26] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *Proc. of the IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011.

5.6 Paper 6: [BMM21]

A COMPARISON OF METHODS FOR OOV-WORD RECOGNITION ON A NEW PUBLIC DATASET

Rudolf A. Braun, Srikanth Madikeri, Petr Motlicek
{rbraun,srikanth.madikeri,petr.motlicek}@idiap.ch

Idiap Research Institute, Martigny, Switzerland

ABSTRACT

A common problem for automatic speech recognition systems is how to recognize words that they did not see during training. Currently there is no established method of evaluating different techniques for tackling this problem.
We propose using the CommonVoice dataset to create test sets for multiple languages which have a high out-of-vocabulary (OOV) ratio relative to a training set and release a new tool for calculating relevant performance metrics. We then evaluate, within the context of a hybrid ASR system, how much better subword models are at recognizing OOVs, and how much benefit one can get from incorporating OOV-word information into an existing system by modifying WFSTs. Additionally, we propose a new method for modifying a subword-based language model so as to better recognize OOV-words. We showcase very large improvements in OOV-word recognition and make both the data and code available.

***Index Terms*—** speech recognition, OOV-word recognition, speech dataset, finite-state transducers

1 Introduction

All languages are constantly evolving and therefore all ASR systems suffer from failing to detect words that were not in their training set (out-of-vocabulary, OOV, words). We focus on weighted finite-state transducer (WFST) based ASR systems with distinct acoustic and language models [1]. In these systems both the language model and lexicon are fixed and encoded as a WFST, this means words that were not part of these systems at training time are impossible to recognize. This has lead to various approaches to modify the WFSTs so that the ASR system can recognize words it had previously no knowledge of [2, 3, 4, 5, 6, 7]. A complication is that typically the lexicon and language model WFSTs will be composed together to create a static decoding graph that can be used repeatedly during decoding. This is a problem because, depending on the use-case, it means we don't have access to the lexicon WFST (L) or the language model WFST (G), and must try and alter the full decoding graph, the HCLG, which is harder.

One workaround is to use a subword-based model, as they can theoretically create any word by outputting a sequence of shorter subword tokens [8, 9, 10]. Another approach is for the language model to contain a $[unk]$ (unknown) token, which has as the pronunciation a phone LM trained on a lexicon of words with low counts, and then to try recover a word from the recognized phone sequence aligned with the $[unk]$ token [11, 12].

Doing graph composition on a client's device can be difficult as it can take a significant amount of time and memory to perform. Therefore it is usually preferred to deploy ASR systems with an already composed decoding graph. If one is willing to redo composition but does not want to retrain the language model modifying the L and G directly is an option. Alternatively, one can avoid having to create the static decoding graph by doing on-the-fly composition, also known as dynamic composition, which is done at runtime [13, 14, 15, 16]. Keeping the G, for example, separate makes it easier to bias the model towards certain words or add new ones to it [2, 3, 4]. However, this approach causes a decrease in decoding speed.

Finally, one can try and modify the static decoding graph (HCLG) [5, 6, 7]. Because of composition and optimization (e.g. determinisation, minimisation, weight-pushing) the initially separated knowledge sources (the lexicon, language model, etc.) are now entangled, making it harder to modify or add new words and pronunciations than when working with the separated L and G.

Many existing papers focusing on OOV recognition used private datasets, which makes results not comparable [2, 5, 8, 11]. Or to create OOVs they keep the top ten thousand (or some other number that is significantly smaller than a real ASR system would use) in the vocabulary and use the rest as OOV words [4, 5, 10, 8]. This evaluation method is problematic because it would overestimate the benefit of using subword-based models as relatively frequent words are not included in the top ten-thousand but the various inflections of them will be seen often during LM training by the subword-based model. This will make it artificially easy to then recover the OOV-word as the subword sequence needed will have a relatively high probability. For the same reason (these artificial OOV-words actually being common when considering inflections) grapheme2phoneme tools will return more accurate pronunciations than would happen with realistic OOV-words.

Therefore, we create reproducible datasets for English and German using CommonVoice[17] where the test set has a large number of realistic OOVs. We release a new tool for calculating error rate metrics, and propose a new metric called "OOV-CER" for measuring OOV-word recognition performance independent of the performance on in-vocabulary words. Using this setup we compare word to subword-based models, check how well OOV recognition works when using a phone LM as the pronunciation for $[unk]$, and compare how effective modifying the L, G and HCLG is. Finally, we propose a new method for modifying the G of a subword-based model to improve performance.
The data and relevant code to modify WFSTs (discussed later) can be found here: github.com/idiap/icassp-oov-recognition.

2 Dataset

The goal is to create a test dataset with a high amount of realistic OOV-words. The approach we use is to have a large vocabulary and

Thanks to Phil Garner for the helpful discussions

then choose utterances from the CommonVoice[17] dataset that contain at least one OOV-word to create the test dataset. The training set is created from the remainder, while excluding those utterances that would lead to a speaker overlap between train and test. For English we used the Librispeech[18] lexicon as the vocabulary, for German we created one by taking the top two hundred thousand words from a text corpus (Europarl). By using large vocabularies gotten from large corpora we ensure that any OOVs will be realistic.

The training and test set size is 280 / 250 and 2.5 / 3 hours for English / German respectively. The OOV ratio is 12.2 / 13.6%. The distribution of the OOV-words is very flat. The English ones tend to be modern words, the top three are “firefox”, “website” and “nudism”. This is because the Librispeech corpus is based on old books, so the vocabulary is old-fashioned. The German OOV-words tend to be compounds words. The English task is harder as the test set text is not only a different domain but also from a different time-period than the vocabulary and text corpus used to train the LM.

3 Metrics

We measure the standard WER (word error rate) and CER (character error rate). Character error rate is a useful measure because if a word has one character wrong that should be a less significant error than if most are incorrect. Additionally, it is useful to know how well OOV-words are recognized independent of performance on in-vocabulary words because OOV-words are more important than for example stop words (“the”, “a”, “and” etc.). This could be done by measuring OOV recall (how many times a OOV-word in the reference is predicted) but this, like WER, treats one or five character mistakes equally. Therefore we developed a new tool for calculating error metrics and propose a new metric called ‘OOV-CER’. The tool is called `texterrors` and is available at github.com/RuABraun/texterrors.

It does character aware alignment of the reference and hypothesis by incorporating the edit distance between words into the substitution cost. The OOV-CER is calculated by getting the index of the OOV-word in the reference, using it to index into the aligned hypothesis and then calculating the edit distance between that word and the reference word. To take into account that a model could output the reference as two separate words, words in the aligned hypothesis that neighbor the index (obtained from where the OOV-word is in the aligned reference) and are aligned with nothing (are insertions) will be pre- or appended to the word in the index.

As an example: The reference is ”words in sentence”, the hypothesis is ”words in sent tense” and the word ”sentence” is the OOV-word and is aligned to ”sent”. To calculate the OOV-CER we first join ”sent” and ”tense”, as the latter is an insertion and aligned next to the OOV-word, and then calculate the CER between ”sentence” and the joined word.

We don’t bother measuring OOV precision as a decrease in performance will already be reflected by an increase in WER/CER. As OOV-words are more important than most in-vocabulary words if the OOV-CER goes down while the WER stays the same after applying some modification to the model, we consider the model as improved.

4 Model biasing mechanisms

A very common use-case is to have some prior knowledge about likely OOV-words, and to want to adjust the model so as to recognize them. In this section, we first review three approaches and introduce a new one. When we mention using a list of OOV-words, we mean a list that has been extracted from the test set relative to our model vocabulary. This is therefore the best case scenario as we know all OOV-words that our model will be asked to recognize. The $[unk]$ symbol is a token that represents an unknown word, jnk is its default pronunciation.

4.1 UNK with non-jnk pronunciation

This method does not actually require any knowledge of possible OOV-words in advance. Rather than having jnk be the pronunciation of the $[unk]$ token, one can replace it with a phone LM trained on the phones from a lexicon of (possible OOV-) words. The LM is inserted in WFST form. Our implementation uses kaldi’s `utils/lang/make_unk_lm.sh`. This allows for an almost arbitrary phone sequence to be recognized.

In figure 1 one can see a simple L. If we wanted to insert just one pronunciation for $[unk]$ we would delete the existing arc from state 0 to 3, then add an arc for each phone in the pronunciation starting from state 0 and ending at state 3. One of these would have $[unk]$ as the output label. To add a phone LM we take an existing WFST over phones P, and connect state 0 in the L to the start state of P with $[unk]$ as the output label, then connect all final states of P to state 3. The connecting arcs will have input disambiguation symbols to ensure the L is still determinisable.

After decoding one then aligns the best-path output lattice to find which phones match to $[unk]$, runs phoneme2grapheme (trained separately), rescores the alternatives with a character LM and gets the best path to get the recovered word. When the training data for the phone LM comes from the lexicon of OOV-words we call this method ’biased unk lm’. To simulate the case when we don’t know what words are OOV we get phones from a lexicon of words with low counts (relative to the text corpus used to train the LM) and call it ‘unk lm’.

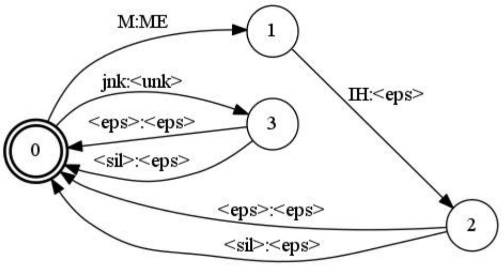


Fig. 1. Simple example of a lexicon WFST (the L).

4.2 Replacing UNK in L and G

This approach assumes one has access to the L and G WFSTs. Using a lexicon of all OOV-words, we add the words and corresponding pronunciations into the L. This is easy to do as the L is unoptimized and we can just add the pronunciations as a sequence of arcs with one of them having the word as the output label. It assumes the new words do not contain any new phones. Then we iterate over the states of the G and replace all arcs with $[unk]$ with multiple arcs keeping the same start and end state, each with one of the OOV-words we want to add as the input and output label. Each arc inherits the $[unk]$ weight plus a penalty of 2.3 (equivalent to multiplying the probability by 0.1). The penalty is because $[unk]$ has a relatively high probability, and we empirically found this to help. This method is called ‘mod L,G’.

4.3 Replacing UNK in HCLG

To replace the `jnk:[unk]` arcs in the HCLG we need an HCL, as the input labels of the HCLG are transition-ids and the states represent different HMM states. We can create an HCL from the lexicon of OOV-words and then do the replacement. For the sake of simplicity our method requires that the HMM topology only has one state. Doing the replace operation makes an additional assumption which constrains the sort of models we can use: By default our models use biphone context dependency, now imagine we inserted the HCL of a word who's pronunciation started with some phone p, the issue is that the input label associated with p should be different depending on what arc came before (i.e. what phone came before) the one we are replacing in the HCLG. But we can't know that at the time of the HCL creation. We get around this problem by using a monophone model. While techniques exist to modify the HCLG of context dependent models [5][7] they are quite complex and we want to test whether using context dependency is even necessary. Due to our LM being trained with the `limit-unk-history` option of *pocolm*, $[unk]$ can only appear at the end of an ngram, so we can just insert the HCL once, and point all arcs matching $[unk]$ to it. The outgoing arcs have the same probability for all histories, as there are no saved histories for $[unk]$. This means the HCLG barely changes in size after the operation. As in 'mod L G' we add in a penalty of 2.3. This method is called 'mod HCLG'.

4.4 Modifying subword G

Trying to modify a word-based model so as to incorporate prior knowledge and better recognize certain (possibly OOV) words is a common focus. However we are not aware of any efforts to try the same with a subword-based model. Since subword-based models can outperform word-based models when there are many OOVs (see section 6), we decided to try incorporate prior knowledge to improve performance even more. We do this by modifying the G (this assumes the G is available separately). We tokenize each OOV-word, and then check if that sequence of subwords exists in the G starting from the backoff state. If it does, we lower the cost (cost because weights are the negative log of the probability) slightly, if it does not we add the necessary arcs with a low cost. The final arc goes to the unigram state of the last subword. This method is called 'mod G'.

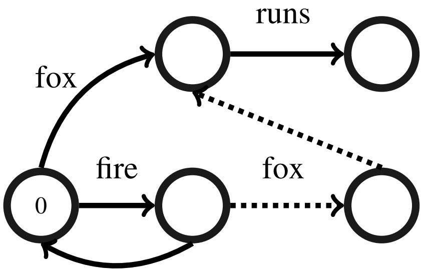


Fig. 2. Illustrative example of how 'mod G' will modify the G by adding new arcs (dashed lines are new arcs) with low costs to increase the odds of recognising certain words. The 0 state is the start state.

In figure 2 a simplified G for illustrative purposes. The 0 state is the start state from which all unigram arcs start. By adding a new (represented by a dashed line) arc 'fox' with a low cost (high probability) from the 'fire' unigram state we lower the total cost of recognizing the subwords 'fire' and 'fox', thereby making it easier for the model to recognize the OOV-word 'firefox'. The alternative, going from the unigram state 'fire' back to state 0 along the backoff arc and then to the unigram state 'fox', would result in a higher total cost for the subword sequence. We also add a back-off arc going to the unigram state 'fox', rather than back to the start state, so that the language model knows that the previous subword was 'fox' which improves performance.

5 Experimental setup

For both languages for the word-based models we train a trigram language model using pocolm, and prune to 3.5 million ngrams. The subword based model uses a five gram pruned to the same number. We use BPE to choose the set of subword tokens and allow 5000 merges. The lexicon of the subword-based model is character based (this performed better than using g2p on the subword tokens). For English the LM training data is the Librispeech text corpus and the we use the 200k lexicon that is part of the corpus, we create pronunciations for OOV words using Phonetisauras[19]. For German we use the Europarl corpus, the vocabulary is the top 200k words, we used espeak-ng for creating the pronunciations.
For training the acoustic model and doing decoding we use kaldi[20]. We follow the standard procedure of getting alignments via HMM-GMM training and then training a TDNNF[21] model with LF-MMI[22] and ivectors. We use biphone context dependency unless indicated otherwise.

6 Results & Discussion

6.1 No prior knowledge

The first case we consider is when no knowledge about potential OOV-words is available. We want to test the assumption that subword-based models do better than word-based, and how well word recovery performs when using the 'unk lm' method. As mentioned previously when using the 'unk lm' method we train once on a lexicon of words with low counts, and once on the lexicon of OOV-words, the latter is 'biased unk lm'. By comparing the two we can test how important it is for the phone LM to be trained on phone sequences that equal the ones seen at test time. The results can be seen in table 1.

		WER	CER	OOV-CER
English	word	36.3	19.7	54.1
	word, unk lm	35.9	18.6	51.8
	word, biased unk lm	35.4	18.7	52.0
	BPE	37.2	19.1	52.1
German	word	29.9	10.2	44.4
	word, unk lm	26.9	9.2	37.2
	word, biased unk lm	25.6	8.8	34.7
	BPE	25.2	8.2	36.0

Table 1. Comparison of word- and subword-based models and OOV recovery using a phone LM when no prior information about OOV-words is known.

Comparing word to subword-based models there is no improvement for English but a significant one for German. These results make sense as the types of OOV-words differ between the two languages. In German a lot of the OOV-words are compounds words, these words can be created by a sequence of subwords which themselves are valid words in the German language and are therefore more likely to be present in ngrams of the ngram language model. In general subword-based LMs benefit from the fact that, unless a

character in a word is very unusual, every word in the training set will be used for training (in segmented form), whereas word LMs will convert all words not part of the vocabulary to $[unk]$.
In the English dataset the OOV-words tend to be completely novel. This means the subword LM is very unlikely to have seen the sequence of subwords, and since there is no natural way to split the OOV-words (because they are not compound words) it is likely that the subwords needed to create the OOV-word will be short (which makes it harder for the language model to make estimates, consider the extreme case of a word being split up into individual characters to understand why), and that no or few n-grams contain these subwords, leading to the language model assigning the subword sequence a low probability.
With the 'unk lm' method one can see an insignificant benefit for English and a noticeable one for German. We decided to test whether the issue was the phone based lexicon for English, and therefore trained a model that used characters as pronunciations. This meant we did not need to do any sort of g2p to get pronunciations for words not in the librespeech lexicon, or do p2g when doing OOV recovery to convert a recognized phone sequence back to letters. We just need to find the characters aligned to $[unk]$ in the decoded lattice. We trained the char LM that is the pronunciation of $[unk]$ on the OOV-word character lexicon. Table 2 shows the results.

		WER	CER	OOV-CER
English	word	36.3	19.7	54.1
	word, unk lm	35.9	18.6	51.8
	word, biased unk lm	35.4	18.7	52.0
	word char	37.0	19.4	53.3
	word char unk lm	36.0	18.8	50.4

Table 2. Comparing OOV recovery with a phone LM to using a model with a character based lexicon, where recovering the word is trivial

The character based model doing OOV recovery does slightly better at recognizing OOV-words, but the WER is still close enough to the phone based baseline model that it is questionable whether the effort is worth it as this is the best case performance since the character LM (used as pronunciation for $[unk]$) was trained on the OOV-word character lexicon. These results show that without having some prior knowledge about the OOV-words the model will encounter, it is very difficult for a hybrid based ASR system to deal with them. In languages with a significant amount of compound words one can use the just described methods to mitigate the amount of errors caused by OOV-words, but the improvement is moderate.

6.2 With prior knowledge

It is a very common use-case to know that certain OOV-words will need to be recognized by a model. We compare three different scenarios: When we have access to the L and G and are willing to redo composition ('mod L,G'), when we don't want to redo composition and therefore modify the HCLG and are willing to accept the constraint of using a monophone model ('mod HCLG'), and when we have a subword-based model and have access to the G and will do composition again ('mod G'). In each case we assume we have a list of OOV-words that we know the model will need to recognize, see section 4 for details on how to incorporate that information. The results are in table 3.

		WER	CER	OOV-CER
English	word	36.3	19.7	54.1
	word mod L,G	24.3	13.8	16.1
	word mono	36.8	19.2	53.2
	word mono mod HCLG	**23.6**	**13.0**	**15.2**
	BPE	37.2	19.1	52.1
	BPE mod G	29.4	15.8	33.4
German	word	29.9	10.2	44.4
	word mod L,G	12.0	**4.9**	4.7
	word mono	30.1	10.4	39.7
	word mod HCLG	**11.8**	5.1	**4.5**
	BPE	25.2	8.2	36.0
	BPE mod G	14.8	5.5	11.1

Table 3. Comparison of the baseline to 'mod L,G', a monophone baseline and 'mod HCLG', the BPE baseline and 'mod G' which modifies the subword-based model.

All methods lead to a very large performance improvement on OOV-words. The fact that the monophone model is so competitive with the biphone baseline supports the modern trend of not using context dependent targets for the acoustic models[23][24], and suggests that these targets are more robust to out-of-domain data (as the OOV-CER is lower). The results also show that using the $[unk]$ probability is a legitimate approach for modeling OOV-words, which makes sense since words that will end up OOV tend to have certain characteristics like being nouns. Adding the penalty of 2.3 to the arcs of each added word improved performance by roughly 10%. While 'mod G' improves the performance of the subword-based model significantly, the modifications for word-based models are better. We believe this is because a lot of OOV-words will be represented by several short subwords, and both their and the pronunciations of the OOV-word (as realized by connecting the pronunciations of the subwords) can be inaccurate, making it hard for the model to recognize the exact sequence of subwords needed to create the OOV-word.

7 Conclusion

We used CommonVoice to create shareable datasets for evaluating OOV-word recognition in English and German. Using a new tool `texterrors` we developed for calculating error metrics, we conducted experiments on OOV recognition performance across two languages in two different scenarios: Without and with prior knowledge. When no prior knowledge is available subword-based models and OOV-word recovery, with a phone LM for $[unk]$, improve results slightly. With prior knowledge we showed several methods to dramatically reduce the error rate on OOV-words. The best approach for dealing with a high OOV-ratio is to use a word-based, context independent model and a modified HCLG. We have shared the data and the code so that others can evaluate their own methods, compare to an existing baseline and build upon our results.

8 Acknowledgements

The work was supported by the European Union's Horizon 2020 Project No. 864702—ATCO2 (Automatic collection and processing of voice data from air-traffic communications), which is a part of the Clean Sky Joint Undertaking.The research is also partially based upon the work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via AFRL Contract FA8650-17-C-9116.

9 References

[1] Mehryar Mohri, Fernando Pereira, and Michael Riley, *Speech Recognition with Weighted Finite-State Transducers*, pp. 559–584, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[2] Petar Aleksic, Mohammadreza Ghodsi, Assaf Michaely, Cyril Allauzen, Keith Hall, Brian Roark, David Rybach, and Pedro Moreno, "Bringing contextual information to google speech recognition," in *INTERSPEECH*, 2015.

[3] Petar Aleksic, Cyril Allauzen, David Elson, Aleks Kracun, Diego Melendo Casado, and Pedro J. Moreno, "Improved recognition of contact names in voice commands," in *ICASSP*, 2015.

[4] J. Novak, N. Minematsu, and K. Hirose, "Dynamic grammars with lookahead composition for wfst-based speech recognition," in *INTERSPEECH*, 2012.

[5] Cyril Allauzen and Michael Riley, "Rapid vocabulary addition to context-dependent decoder graphs," in *INTERSPEECH*, 2015.

[6] Johan Schalkwyk, I. Hetherington, and Ezra Story, "Speech recognition with dynamic grammars using finite-state transducers," in *INTERSPEECH*, 2003.

[7] Anna Bulusheva, Alexander Zatvornitsky, and Maxim Korenevsky, "An efficient method for vocabulary addition to wfst graphs," in *TSD*, 2016.

[8] Samuel Thomas, Kartik Audhkhasi, Zoltán Tüske, Yinghui Huang, and Michael Picheny, "Detection and recovery of oovs for improved english broadcast news captioning," 2019, pp. 2973–2977.

[9] Y. He, B. Hutchinson, P. Baumann, M. Ostendorf, E. Fosler-Lussier, and J. Pierrehumbert, "Subword-based modeling for handling oov words in keyword spotting," in *ICASSP*, 2014.

[10] Maximilian Bisani and Hermann Ney, "Open vocabulary speech recognition with flat hybrid models," 2005, pp. 725–728.

[11] Issam Bazzi, *Modelling Out-of-Vocabulary Words for Robust Speech Recognition*, Ph.D. thesis, 2002.

[12] Asadullah Tanel Alumäe, Ottokar Tilk, "Advanced rich transcription system for estonian speech," 2019.

[13] Cyril Allauzen, Michael Riley, and Johan Schalkwyk, "A generalized composition algorithm for weighted finite-state transducers," in *INTERSPEECH*, 2009.

[14] Octavian Cheng, John Dines, and Mathew Magimai.-Doss, "A generalized dynamic composition algorithm of weighted finite state transducers for large vocabulary speech recognition," Tech. Rep., IDIAP, 2006.

[15] Cyril Allauzen and Michael Riley, "Pre-initialized composition for large-vocabulary speech recognition," in *INTERSPEECH*, 2013.

[16] J. Liu, Jiedan Zhu, Vishal Kathuria, and Fuchun Peng, "Efficient dynamic wfst decoding for personalized language models," *ArXiv*, vol. abs/1910.10670, 2019.

[17] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber, "Common voice: A massively-multilingual speech corpus," 2020.

[18] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *ICASSP*, 2015.

[19] J. Novak, N. Minematsu, and K. Hirose, "Wfst-based grapheme-to-phoneme conversion: Open source tools for alignment, model-building and decoding," in *FSMNLP*, 2012.

[20] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. Dec. 2011, IEEE Signal Processing Society, IEEE Catalog No.: CFP11SRW-USB.

[21] Daniel Povey, Gaofeng Cheng, Yiming Wang, Ke Li, Hainan Xu, Mahsa Yarmohammadi, and Sanjeev Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks," in *INTERSPEECH*, 2018.

[22] D. Povey, Vijayaditya Peddinti, D. Galvez, Pegah Ghahremani, Vimal Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for asr based on lattice-free mmi," in *INTERSPEECH*, 2016.

[23] Vineel Pratap, Awni Hannun, Qiantong Xu, Jeff Cai, Jacob Kahn, Gabriel Synnaeve, Vitaliy Liptchinsky, and Ronan Collobert, "Wav2letter++: A fast open-source speech recognition system," in *ICASSP*, 2019.

[24] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai, "ESPnet: End-to-end speech processing toolkit," in *INTERSPEECH*, 2018.

5.7 Paper 7: [MVG10]

English Spoken Term Detection in Multilingual Recordings

Petr Motlicek, Fabio Valente, Philip N. Garner

Idiap Research Institute, Martigny, Switzerland
petr.motlicek@idiap.ch, fabio.valente@idiap.ch, pgarner@idiap.ch

Abstract

This paper investigates the automatic detection of English spoken terms in a multi-language scenario over real lecture recordings. Spoken Term Detection (STD) is based on an LVCSR where the output is represented in the form of word lattices. The lattices are then used to search the required terms. Processed lectures are mainly composed of English, French and Italian recordings where the language can also change within one recording. Therefore, the English STD system uses an Out-Of-Language (OOL) detection module to filter out non-English input segments. OOL detection is evaluated w.r.t. various confidence measures estimated from word lattices. Experimental studies of OOL detection followed by English STD are performed on several hours of multilingual recordings. Significant improvement of OOL+STD over a stand-alone STD system is achieved (relatively more than 50% in EER). Finally, an additional modality (text slides in the form of PowerPoint presentations) is exploited to improve STD.

Index Terms: Spoken Term Detection (STD), LVCSR, Confidence Measure (CM), Out-Of-Language (OOL) detection

1. Introduction

A large increase in the amount of spoken recordings requires automatic indexation and search in this data. A potential solution is a Spoken Term Detection (STD) system[1] which would be able to quickly detect a word or phrase in large archives of unconstrained speech recordings (e.g. lecture recordings, telephone conversations, ...). A common approach is to split the task into two stages. Firstly, a Large Vocabulary Continuous Speech Recognition (LVCSR) system is used to generate a word or phone lattice. Secondly, lattice search is performed to determine likely occurrences of the search terms. STD systems based on word lattices provide significantly better performance than those based on phoneme lattices (e.g., [1]). Word lattices can be associated with a Confidence Measure (CM) for each word. Traditionally, forward-backward re-estimation is used to represent a confidence using word posterior probability conditioned on the entire utterance. Although such an STD system does not deal with Out-Of-Vocabulary (OOV) words, the problem can be solved by taking into account phone recognition lattices usually generated by the same LVCSR system.

In this paper, we present experimental results with an LVCSR-STD system performing automatic indexation of real lecture recordings provided by Klewel[2] to be eventually implemented into a conference webcasting system. Most of the Klewel lecture talks are recorded in west Switzerland. Speech recordings are mostly uttered in English (usually by non-native speakers), however, some recordings are partially (e.g. at the beginning of the talk), or fully uttered in French or Italian. Blindly applying an English STD system for automatically indexing English pronunciations in such multilingual recordings would lead to a significant decrease of overall STD performance since the system would be employed on “inappropriate” speech input (i.e., speech pronounced in different (alien) languages whose words do not appear in the LVCSR dictionary). The amount of detected False Alarms (FAs) of searched terms would significantly increase. These FAs could potentially be reduced by modifying an operating point of the STD system, but this would lead (directly) to an increase of missed terms.

A straightforward solution is to employ a language identification module. However, such a system would have to exploit the knowledge of other (non-target) languages. In order to keep the entire STD system relatively simple and independent of any non-target language, an OOL detection module is an ideal solution. Such a module exploits only the information stored in the same LVCSR word lattices used for search of the spoken terms. Similar approach can possibly be applied to reduce false detections due to dialect variations of the target language which usually have a severe impact on the performance of speech systems [2].

The paper is organized as follows: Sect. 2 and Sect. 3 describe respectively STD task and the STD system used in our studies. Experiments carried out to improve the STD system and achieved results are given in Sect. 4. Sect. 5 concludes the paper.

2. STD task

2.1. Test data

The study is carried out on the 16 kHz audio lecture recordings (supplemented with video and text) provided by Klewel[2]. In total, 9 hours of recordings pronounced in English, French and Italian languages were used. This data was first transcribed according to the input language and then used for evaluation of the OOL detection module. Then, over one hour of English data (from 9 hours of multilingual speech) was selected for STD evaluations and carefully manually annotated. In order to jointly evaluate STD and OOL modules, an additional two hours of French and Italian recordings were used together with one hour of English data. All audio recordings were automatically segmented using a state-of-the-art Multi-Layer Perceptron (MLP) based speech/non-speech detector [3].

In addition, to evaluate a stand-alone STD English system on a standard database, 3 hours of a two channel 8 kHz CTS English development corpus distributed by NIST for the 2006 STD task was used[1].

This work was partially supported by the Swiss National Center of Competence in Research (NCCR) on “Interactive Multi-modal Information Management (IM)2”; and by the European commission $7th$ Framework Programme (FP7) ICT project TA2.

[1] NIST STD06 Eval, http://www.itl.nist.gov/iad/mig//tests/std

[2] http://www.klewel.com

2.2. Evaluation metric

Since STD is a detection task, performance can be characterized by Detection Error Tradeoff (DET) curves of miss (P_{miss}) versus false alarm (P_{fa}) probabilities [4]. In addition, we also present Equal Error Rates (EERs), a one number metric often used to optimize the system performance. Besides DET and EERs, we use the evaluation measure defined by NIST 2006 STD: Maximum Term-Weighted Value (MTWV) [5].

3. STD system

To perform the search of selected spoken terms in lecture audio recordings, the recordings are first pre-processed using the LVCSR system that produces word recognition lattices. The word lattices are then converted into a candidate term index accompanied with times and detection scores. The detection scores are represented by the word posterior probabilities, estimated from the lattices using the forward-backward re-estimation algorithm [6], and defined as:

$$P(W_i; t_s, t_e) = \sum_Q P(W_i^j; t_s, t_e | x_{t_s}^{t_e}), \qquad (1)$$

where W_i is the hypothesized word identity spanning the time interval $t \in (t_s, t_e)$. t_s and t_e denote the start and end time interval, respectively. j denotes the occurrence of word W_i in the lattice. $x_{t_s}^{t_e}$ denotes the corresponding partition of the input speech (the observation feature sequence). Q represents a set of all word hypotheses sequences in the lattice that contain the hypothesized word W_i in $t \in (t_s, t_e)$.

3.1. LVCSR system

To achieve robust hypotheses outputs, a 3-pass LVCSR system is employed, based on various acoustic models trained on different audio data (no Klewel recordings used for training). The system achieving the best recognition performance is then selected to be used in the subsequent STD experiments. More specifically, an LVCSR based on the 8 kHz Conversational Telephone Speech (CTS) system derived from AMI[DA][3] LVCSR [7] is used. In the first pass, PLP features are exploited and HMMs are trained using a Minimum Phone Error (MPE) procedure. In the second pass, Vocal Tract Length Normalization takes place together with HLDA, MPE and Speaker Adaptive Training (SAT). In the third pass, posterior-based speech features estimated using a neural network system replace PLPs. For the decoding, a 50k dictionary is used together with a 3-gram Language Model (LM).

In the second potential system, acoustic models trained on 16 kHz Individual Headset Microphone (IHM) recordings from several meeting corpora (ICSI, NIST, AMI) are employed, replacing CTS models. In the third case, Multiple Distant Microphone (MDM) instead of IHM recordings are used to train acoustic models. In both (IHM, MDM) cases, discriminative training in 3-pass system, similar to the previous AMI CTS system, is employed.

To select the most suitable LVCSR setting in the following STD studies, we evaluate the three systems on 1 hour of manually annotated Klewel English lectures. Overall, the best ASR performance measured in terms of Word Error Rates (WERs) is achieved for the LVCSR system trained on 16 kHz IHM meeting recordings (WER = 28.9%). LVCSR systems trained on 16 kHz MDM and 8 kHz CTS acoustic models perform around 4% and 6% worse, respectively. Therefore, 16 kHz IHM LVCSR is selected for subsequent STD studies.

3.2. Evaluation of stand-alone STD system

First, the LVCSR STD system is evaluated on 3 hours of 8 kHz CTS English development database. The automatically segmented speech recordings are processed by the AMIDA LVCSR system employing CTS acoustic models with a 50k dictionary. The generated bigram lattices are subsequently expanded with a trigram language model. For evaluation, 550 English search terms are randomly selected from the STD06 dry-run list. The achieved STD performance is compared to the baseline system described in [8]. The EER of the baseline system is about 10.1%. The presented STD built on 3-pass LVCSR gives about 20% relative improvement.

For automatic indexing of Klewel lecture recordings, an STD system based on word lattices generated using 16 kHz IHM acoustic models is chosen, since the best ASR performance is achieved with such a system. Word recognition lattices are generated in the third pass using HTK (HDecode) with bi-gram language model. The list of English spoken terms consists of 312 items. The terms are selected manually from the available annotations (in a random fashion over all recordings based only on a potential interest of Klewel end-users). The list of terms is then transformed into a format following NIST 2006 STD evaluations. The EER achieved on 3 hours of Klewel multilingual recordings is about 8.1%, as shown in Tab. 2.

4. Improving STD by detecting OOL segments

Although the English STD system performs reasonably well, while having at the input (unrestricted) multilingual recordings, other improvements can be obtained by detecting OOL segments. The OOL module can be thought of as a probabilistic model that assigns a probability of each processed input segment given the language used in the segment.

4.1. OOL module

The OOL detection used extracts a confidence score of the processed input speech using several Confidence Measures (CMs) [9]. These CMs are derived from word LVCSR lattices. More specifically, we studied these CMs:

- C_{mean} – Probabilities of all hypotheses for the word W_i recognized in the 1-best output, spanning time interval $t \in (t_s, t_e)$, are summed and normalized [10]:

$$C_{mean} = \frac{\sum_{t=t_s}^{t_e} P(W_i \mid t)}{1 + \alpha(t_e - t_s - 1)}. \qquad (2)$$

α is a constant between 0 and 1.

C_{max} – The best case probability for a hypothesized word W_i (also found in the 1-best output) occurring in a certain period of time $t \in (t_s, t_e)$ is returned [10]:

$$C_{max} = \max_{t \in (t_s, t_e)} P(W_i \mid t). \qquad (3)$$

- $H\big(W \mid t_{t_s}^{t_e}\big)$ – Amount of uncertainty of recognized words measured using Entropy information criteria for the given time interval $t \in (t_s, t_e)$:

$$H\big(W \mid t_{t_s}^{t_e}\big) = \frac{\sum_{t=t_s}^{t_e} \sum_i \frac{1}{P(W_i|t)} log_2\big(P(W_i \mid t)\big)}{1 + \alpha(t_e - t_s - 1)}. \qquad (4)$$

[3] http://www.amiproject.org

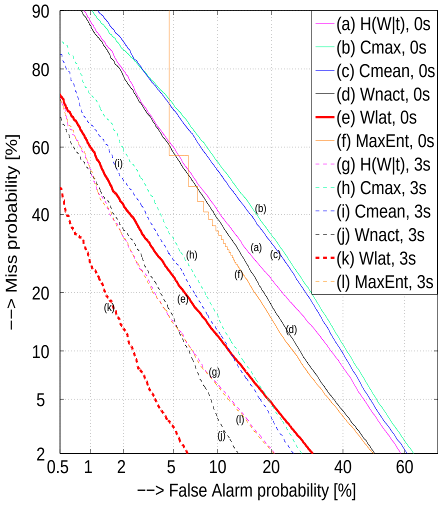


Figure 1: *DET plot – OOL detection using different CMs for temporal context equal to* 0 *sec. and* 3 *sec.*

OOL - EER					
Len [s]	C_{mean}	C_{max}	$H(W \mid t_{t_s}^{t_e})$	W_{lat}	W_{nact}
0	24.9%	25.6%	21.4%	**10.9%**	19.0%
3	11.2%	11.8	8.0%	**4.1%**	7.4%
120	3.6%	3.9%	2.6%	**1.4%**	2.6%

Table 1: *OOL – EER [%] performances achieved on Klewel lecture recordings for different CMs and various temporal context.*

- W_{lat} – Word lattice width - a simple measure provided by counting the number of active arcs from the recognition lattice determines the amount of uncertainty in the LVCSR system at the given time instance $t = t_n$.

- W_{nact} – Number of active and unique words at the given time instance $t = t_n$ is counted and also used as an OOL confidence score.

OOL detection is tested directly on the target Klewel evaluation data. In particular, 9 hours of recordings (3 hours from each of English, Italian and French language) are used. The derived OOL CMs, described in Sec. 4.1, are further post-processed to incorporate a temporal context. This has been shown to significantly improve the detection performance. In case of unconstrained length of processed speech segments, the optimal length of the temporal filter was found to be about 3 sec. [9]. We also experimented with higher lengths (up to 120 sec.) of the filter, since the language usually does not change often in the processed recordings. However, this may cause significant degradation of OOL detection when such a temporal filter were applied on short speech segments, as shown in [9].

Achieved detection performance is shown in the form of DET curves and EERs in Fig. 1 and Tab. 1, respectively. W_{lat} as a confidence score significantly outperforms other CMs used for OOL detection. Additional experiments to fuse all individual CMs using a Maximum Entropy (MaxEnt) technique do not bring any improvements (see Fig. 1). This is probably caused by employing very different data to train the MaxEnt classifier.

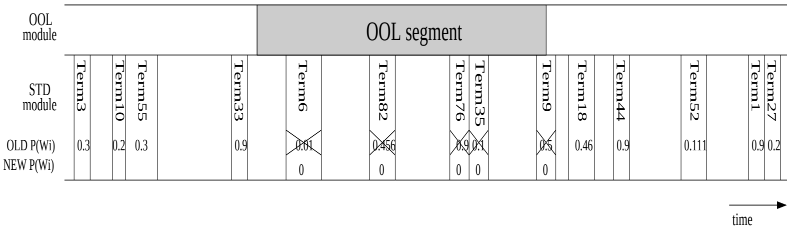


Figure 2: *Combination of OOL and STD modules: STD detection scores are set to zero if detected in speech segments marked as OOL.*

STD						
	OOL - W_{lat}		OOL - no		OOL - manual	
Len [s]	EER	MTWV	EER	MTWV	EER	MTWV
0	5.9%	0.70	8.1%	0.64	3.5%	0.82
3	4.0%	0.78				
120	3.6%	0.81				

Table 2: *STD – EER [%] performances achieved on Klewel lecture recordings w.r.t. OOL detection module. Len denotes length of the temporal filter of the OOL detection module. OOL-W_{lat}, OOL-manual and OOL-no denote OOL detection based on W_{lat} CM, OOL detection taken from manual annotations and the STD system without OOL detection module, respectively.*

4.2. Exploiting OOL in STD system

The OOL detection module is applied in the STD system to automatically remove input speech segments pronounced in non-target languages. Therefore, false alarm terms caused by processing non-English speech segments will potentially be removed in an optimal way (i.e., without any effect on correctly detected terms in English segments).

More specifically, the confidence scores of those terms (already detected by STD system) which correspond to speech segments classified to be OOL segments are set to zero, as graphically shown in Fig 2. In order to “hard threshold” STD detection scores using the OOL detection module, an OOL detection threshold needs to be introduced. In our studies, an optimal threshold is found on development data. A development set comprising of 30 min. of audio recordings uttered in Czech and German languages (i.e., different to French and Italian) as well as in English is used to tune the operating point of OOL detection module [9]. The threshold corresponding to EER is selected as the operating point of the OOL detection module.

Experimental results of the English STD system, in terms of EERs and MTWVs, achieved on 3 hours of multilingual Klewel lecture recordings are given in Tab. 2. Graphical representation in terms of DET curves is shown in Fig 3. Since the best automatic OOL detection performace is achieved with W_{lat} CM, that system is exploited in STD experiments. As seen in Tab. 2, the temporal filter of the OOL detection module with a length of 3 sec. gives performance close to the STD system with manually classified OOL speech segments.

4.3. Exploiting prior information from other modality

Many Klewel lecture audio recordings are supplemented with corresponding slide (PowerPoint) presentations. Therefore, we attempted to exploit this modality in our STD experiments.

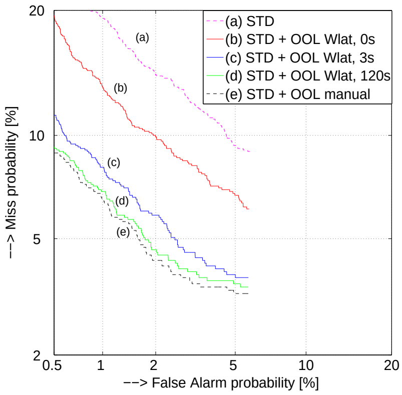


Figure 3: *DET plot – STD on Klewel multilingual recordings.*

STD			
slide	OOL	EER	MTWV
no	no	5.3%	0.74
yes	no	4.5%	0.76
yes	Wlat, 3s	2.0%	0.80
yes	manual	1.6%	0.83

Table 3: *STD – EER [%] performances achieved on a subset of Klewel lecture recordings when additional modality is exploited. c was chosen to be equal to* 50*.*

More specifically, word posterior probabilities $P(W_i; t_s, t_e)$ of searched terms are modified using a prior which represents a relevance of a term to the topic (given by corresponding text slides). The prior is introduced by a multiplicative constant c:

$$\begin{array}{ll} P_{new} = cP_{old}, & if \quad c <= 1/P_{old}, \\ P_{new} = 1, & otherwise. \end{array} \qquad (5)$$

The experiments are run on a subset ($\sim 1/3$) of the multilingual lecture recordings (those supplemented with text slides). First, for each lecture recording, a new list of terms (which is a subset of original 312 searched terms) is automatically generated based on the occurrence of searched terms in the text of corresponding PowerPoint slides. Since no time allocation of the individual slides and their precise alignment with the audio segments of a lecture is available (only the general lecture number assignation), no precise temporal information is employed. Then, posterior probabilities P_{old} (initially estimated from the LVCSR lattices) associated with search terms occurring in the new list of a given lecture are updated according to Eq. 5.

Fig. 4 graphically shows a dependence of EER on varying c for two STD systems (without and with application of the OOL detection module). c varied from 10^{-4} to 10^{3}. Corresponding MTWV values are given in Tab. 3. Although a very simple model is used, which takes into account neither time allocation of searched terms nor quantity of their occurrence in the corresponding slides of each lecture, a relative EER improvement of about 15% is achieved (in both cases with and without the OOL detection module).

5. Discussions and conclusions

This paper summarizes experimental results achieved with an English STD system on Klewel lecture recordings. Due to the unconstrained multilingual input, the system is augmented with an OOL detection module which assigns to each input segment (e.g. frame) a probability given the language used in the segment. Such a module performs as a binary classifier (target-*English* / non-target-*any* language). An OOL detection module can use different lengths of temporal context, which has a significant effect on performance of the subsequent STD system.

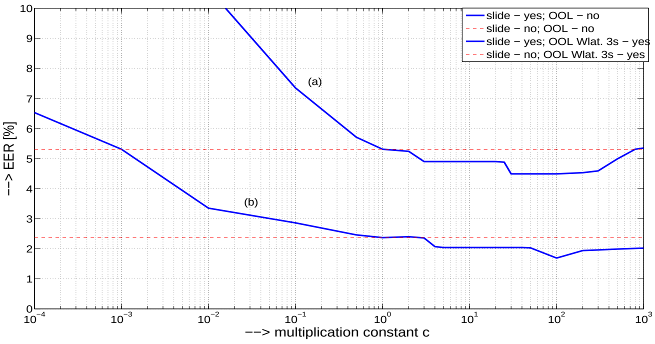


Figure 4: *Overall EERs of STD on the subset of Klewel multilingual recordings when additional prior information is exploited: (a) STD system without OOL module, (b) STD system with OOL module.*

STD performance is measured using several criteria (DET curves, EER, MTWV values) on 3 hours of multilingual recordings. Incorporation of the OOL detection module (with 3 sec. long temporal filter) into the STD system increases EER performance relatively by more than 50%.

We also experimented with an additional source of information available from associated text slides on a subset of Klewel recordings. Posterior probabilities (initially estimated from the LVCSR lattices) of those spoken terms which are detected in the corresponding slides of a given lecture recording are modified by a multiplicative constant. A relative improvement of $\sim$ 15% in STD EER was found.

6. References

[1] D. Vergyri et al., "The SRI/OGI 2006 Spoken Term Detection System", *in Proc. of Interspeech*, pp. 2393-2396, Belgium, 2007.

[2] M. Mehrabani, H. Boril and J. Hansen, "Dialect Distance Assessment Method based on Comparison of Pitch Pattern Statistical Models", *in Proc. of ICASSP*, pp. 5158-5161, Dallas, USA, 2010.

[3] J. Dines, J. Vepa, and T. Hain, "The segmentation of multichannel meeting recordings for automatic speech recognition", *in Proc. of the ICSLP*, pages 1213-1216, Pittsburgh, USA, 2006.

[4] A. Martin, G. Doddington, T. Kamm, M. Ordowski, M. Przybocki, "The DET curve in assessment of detection task performance", *in Proc. of Eurospeech*, pp. 1895-1898, Greece, 1997.

[5] NIST Spoken Term Detection (STD) 2006 Evaluation Plan, <http://www.itl.nist.gov/iad/mig//tests/std/2006/docs/std06-evalplan-v10.pdf>

[6] G. Evermann and P. Woodland. "Large Vocabulary Decoding and Confidence Estimation using Word Phoneme Accuracy Posterior Probabilities", *in Proc. of ICASSP*, pp. 2366-2369, Turkey, 2000.

[7] T. Hain, et al, "The AMI System for the Transcription of Speech in Meetings", *in Proc. of ICASSP*, pp. 357-360, Hawaii, USA, 2007.

[8] I. Szoke et al., "BUT System for NIST Spoken Term Detection 2006 - English", *in Proc. of NIST Spoken Term Detection Workshop (STD 2006)*, pp. 15, Washington D.C., USA, 2006.

[9] P. Motlicek, "Automatic Out-of-Language Detection based on Confidence Measures derived from LVCSR Word and Phone Lattices", *in Proc. of Interspeech*, Brighton, England, 2009.

[10] F. Wessel, R. Schluter, K. Macherey and H. Ney, "Confidence measures for large vocabulary continuous speech recognition", *in IEEE Trans. Speech and Audio Processing*, vol. 9, no. 3, pp. 288-298, 2001.

5.8 Paper 8: [Oua+17]

A CONTEXT-AWARE SPEECH RECOGNITION AND UNDERSTANDING SYSTEM FOR AIR TRAFFIC CONTROL DOMAIN

Youssef Oualil[1], *Dietrich Klakow*[1], *György Szaszák*[1],
Ajay Srinivasamurthy[3], *Hartmut Helmke*[2], *Petr Motlicek*[3]

[1]Spoken Language Systems Group (LSV), Saarland University (UdS), Saarbrücken, Germany
[2]Institute for Flight Guidance, German Aerospace Center (DLR), Braunschweig, Germany
[3]Idiap Research Institute, Martigny, Switzerland

`firstname.lastname@{lsv.uni-saarland.de, dlr.de, idiap.ch}`

ABSTRACT

Automatic Speech Recognition and Understanding (ASRU) systems can generally use temporal and situational context information to improve their performance for a given task. This is typically done by rescoring the ASR hypotheses or by dynamically adapting the ASR models. For some domains such as Air Traffic Control (ATC), this context information can be however, small in size, partial and available only as abstract concepts (e.g. airline codes), which are difficult to map into full possible spoken sentences to perform rescoring or adaptation. This paper presents a multi-modal ASRU system, which dynamically integrates partial temporal and situational ATC context information to improve its performance. This is done either by 1) extracting word sequences which carry relevant ATC information from ASR N-best lists and then perform a context-based rescoring on the extracted ATC segments or 2) by a partial adaptation of the language model. Experiments conducted on 4 hours of test data from Prague and Vienna approach showed a relative reduction of the ATC command error rate metric by 30% to 50%.

Index Terms— Automatic speech recognition, context-aware systems, air traffic control, spoken language understanding.

1. INTRODUCTION

Automatic Speech Recognition and Understanding (ASRU) applications can generally benefit from the presence of task-related situational and temporal context (prior) information to improve their performance [1]. This can be done either by 1) refining the ASRU models, such as adapting the acoustic model to new acoustic conditions or adapting the Language Model (LM) to a new domain, or 2) by rescoring the ASR hypotheses using a domain-dependent model. Early usage of situational context goes back to Young et al.'s works [2, 3], who used sets of contextual constraints to generate several grammars for different contexts. Fügen et al. [4] used a dialogue-based context to update a Recursive Transition Network (RTN) to improve ASR quality of a dialogue system. Everitt et al. [5] proposed a dialogue system for gyms, which, based on the exercise routine, would switch its ASR component between pre-existing grammars tailored to different sports equipments.

While there is no doubt that context can significantly improve ASRU performance, the information it provides however, can be small in size, time-varying, partial and available only as machine-generated abstract representations (e.g. airline codes on a radar screen), which are difficult to map back into full possible spoken sentences to perform rescoring or adaptation. In particular, in order to manage a given airspace, Air Traffic Controllers (ATCOs) issue verbal commands to the pilots by interpreting and relying on 1) situational context acquired through multiple modalities such as, radar derived aircraft state vectors comprising position, speed, altitude, etc., as well as 2) temporal context given by the sequence of previously issued commands. Furthermore, verbal communication is the primary mode of communication between agents operating in the ATC domain, which inspires many ASRU-based applications to enhance the ATC technologies. The designed ASRU systems can also benefit from the same context information used by ATCOs. Shore et al. [6] investigated this idea using lattice rescoring on a small Context Free Grammar (CFG)-based simulated ATC setup, whereas Schmidt et al. [7] proposed a dynamic finite state transducer adaptation of a CFG-based LM. As an alternative to CFG solutions, we have recently proposed a Levenshtein-based context integration approach combined with a Statistical Language Model (SLM) [8]. More details about ASRU for ATC are presented in Section 2.

This paper extends and generalizes the work presented in [7, 8] in different directions. That is, 1) in addition to situational context, we propose a new model that also integrates temporal context (history of spoken commands) (Section 3). Then, 2) we combine the two types of context in a generalization of [8] using N-best lists (Section 4). Finally, 3) contrary to [7, 8], which evaluated their systems on data collected from a simulator of Düsseldorf airport, this paper evaluates the system on 4 hours of data collected from ATCOs performing their daily tasks in Vienna and Prague airports (Section 6). The obtained results show that the proposed context-aware ASRU system reduces the ATC Command Error Rate (CmdER) metric by 30% to 50% compared to a standard ASRU system.

2. ASRU SYSTEMS FOR ATC DOMAIN

2.1. Air Traffic Control Assistance Systems

The task of air traffic control aims at maintaining a safe, orderly and expeditious flow of air traffic. ATCOs apply strict separation rules to direct aircraft safely and efficiently, both in their respective airspace sector and on the ground. Since controllers have a significant responsibility and can face high workloads in busy sectors, different planning systems have been proposed to assist them in managing the airspace such as, the Arrival Manager (AMAN). These systems mainly suggest an optimal sequence of commands (command advisories), which are then issued in verbal radio communication from the controller to the aircraft pilots.

This work was supported by the MALORCA project (Grant Agreement No. 698824), funded by SESAR Joint Undertaking, under EU Horizon 2020.

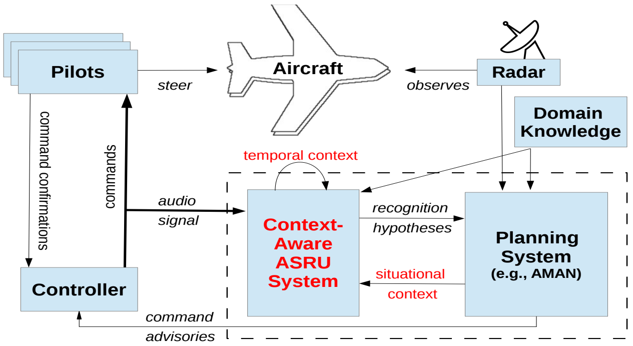


Fig. 1. Schematic view of an ASRU-based ATC system.

2.2. AcListant®: Active Listening Assistance System

For different reasons such as, emergency or weather conditions, the controller may deviate from the advisory commands proposed by the assistance system. The latter reacts slowly to such deviations and may require the controller to enter the issued commands via mouse/keyboard. Thus, indirectly increasing the workload that they were mainly designed to reduce. As a solution to this problem, we have recently proposed the AcListant®[1] [9] system, which extends the planner to include a background ASRU system, ideally replacing the mouse/keyboard feedback. Conversely, ASRU can also benefit from the context information used by the assistant system [8, 10] to improve its performance. We will refer to it as Assistance-based ASRU (ABSRU) system in the rest of this paper. Fig. 1 shows the information flow in an ASRU-based assistance system.

2.3. From AcListant® To MALORCA

Although the AcListant® system achieved a good performance in a simulator of Düsseldorf airport, the cost of transferring such system from the laboratory to real ops-rooms is very significant. Each model in the ABSRU system must by manually adapted to the linguistic and acoustic features of the new environment, which are due to new local conditions such as, noise conditions, different accents, speaking styles, deviations from standard phraseology [11], etc. Therefore, the MALORCA[2] project is proposed as a generalization of AcListant® that aims at developing a general, cheap and effective solution to automate the re-learning, adaptation and customization process to new environments. This will be done by taking advantage of the large amount of un-transcribed speech data available on a daily basis in the new ATC environment, which can be used in un/semi-supervised learning approaches to automatically adapt the ABSRU models to the respective environment. The work presented in this paper describes the basic and general ABSRU systems, which will be used as initial points in the bootstrap automatic adaptation cycle for Vienna and Prague airports, respectively.

3. ATC CONTEXT-BASED RESCORING

This section introduces the different types of context we consider and the mathematical models we designed to integrate them into an ASRU system. Then, we show how these different models can be combined in a unifying framework.

3.1. Situational Context Information

An ATC assistance system bases its proposed command sequence on the state of a given airspace sector. This state is primarily derived from radar information about the current situation of the airspace and aviation domain knowledge. This is done by forming a search space of all physically possible commands in the current airspace situation in a first step, and then extracting the advisory sequence of commands, shown to ATCOs, by optimizing a set of ATC criteria. The formed search space summarizes the current situation in the airspace. Thus, we will refer to it as **situational context**. For an ASRU system, this context can be seen as a command-level search space, which is 1) dynamic, i.e. changes every few seconds, 2) small in size, i.e. few hundred/thousand of commands, and 3) available only as partial standardized ICAO phraseology concepts [11] (see example Table 1). In particular, a situational context information contains an aircraft callsign (e.g. AFR2A ≅ *air france two alpha*) followed by a command type to execute and a command value to achieve (e.g. REDUCE 220 ≅ *reduce speed two two zero knots*).

Callsign	Command Type	Value
AFR2A	REDUCE	220
DLH9000	DESCEND	120
BER256	RATE_OF_DESCENT	3000
KLM23RV	TURN_LEFT_HEADING	80

Table 1. Excerpt from situational context information generated by a planning system. It shows an ICAO abstraction of four different actions that can be issued by the controller to an aircraft.

Given the spoken language variability, it is very difficult to build the word-level context space, which maps each command in the context into the set of all possible spoken realizations of that command, which can be issued by an ATCO to an aircraft pilot. Furthermore, such process should be very fast given that the situational context changes every few seconds. As a result, performing the standard lattice rescoring or LM adaptation is not feasible in this case. The next section introduces a partial rescoring approach, which considers only the ATC segments in the recognized hypotheses.

3.2. Situational Context-based Rescoring (SCR)

The situational context model considers the context information as an ASRU search space for ATC concepts. That is, it only targets sequence of words that carry some ATC information in the recognized hypotheses. This partial rescoring approach follows these steps:

Step 1) Sequence Labeling: This step takes the raw ASR hypothesis as input and automatically **detects and extracts** the ATC concepts that it carries. For instance, the hypothesis "*air france two alpha hello reduce speed two three zero knots*" is mapped to "`<callsign>` *air france two alpha* `</callsign>` *hello* `<command=reduce>` *reduce speed* `<speed>` *two three zero* `</speed>` *knots* `</command>`". This step directly puts the focus on the ATC information carried by the ASR hypotheses, which is our primary target, and ignores the rest. Our experiments use a CFG-based token tagger similar to the one used in [7, 8].

Step 2) Context-to-Word Mapping: The partial rescoring approach turns the problem of generating full spoken sentences (realizations) of the context into generating realization of short segments, which can be extracted by the sequence labeler in the previous step. For instance, instead of generating the full realization of the command "AFR2A REDUCE 250", we only need to generate context-to-word mapping for the callsign "AFR2A" and the speed value "250".

[1] AcListant®: http://www.AcListant.de

[2] MALORCA: MAchine Learning Of speech Recognition models for Controller Assistance: http://www.malorca-project.de

Step 3) Situational Context-based Rescoring: We use here a Weighted Levenshtein Distance (WLD) to rescore the ATC segments extracted from the ASR hypotheses in Step 1, in the search space formed by all verbalized context segments from Step 2. More details about the WLD can be found in [8].

Formally, let $A = \{A_{cs}, \{A^{cs}_{com}\}_{com}\}$ be the ATC segments extracted from the ASR hypothesis using sequence labeling as described in Step 1. We assume that each hypothesis contains (at most) a single callsign A_{cs} in addition to one or multiple issued commands $\{A^{cs}_{com}\}_{com}$. Similarly, let $\mathcal{C} = \cup_{cs}\{(C_{cs}, \{C^{cs}_{com}\}_{com})\}$ be the set of all possible context-based ground truths resulting from the context-to-word mapping described in Step 2. This set consists of all callsigns in the context and the ATC commands applicable to them. The situational context-based rescoring extracts the "corrected" ATC segments $H = \{H_{cs}, \{H^{cs}_{com}\}_{com}\}$ according to

$$\begin{aligned} H &= \underset{C\in\mathcal{C}}{\operatorname{argmin}}\{WLD(A, C)\} \qquad (1) \\ &= \underset{C\in\mathcal{C}}{\operatorname{argmin}}\{WLD(A_{cs}, C_{cs}) + \sum_{A_k\in\{A^{cs}_{com}\}} \underset{C_j\in\{C^{cs}_{com}\}}{\operatorname{argmin}} WLD(A_k, C_j)\} \end{aligned}$$

More details about the WLD and the situational context-based rescoring can be found in [8].

3.3. Temporal Context-based Rescoring (TCR)

Air traffic control assistance systems typically use the radar information to generate the situational context. The resulting command advisories are generated through a deterministic optimization process, which takes into account a number of physical and local constraints about the operating airport. These constraints include waypoints, which play the role of "markers" in the airspace, location of the runways for landing and departure, the landscape surrounding the airport (see, mountains, etc), to name a few. Due to these constraints, a number of pre-defined trajectories and landing patterns are frequently generated to guide aircraft from their current location to the runways. For instance, most landing aircrafts will receive a confirmation of identification as first command, and a handover as last command. In particular, once an aircraft enters the controlled airspace, the generated landing sequence for this aircraft is expected to be closely similar to the ones generated for previous aircraft that entered that airspace at close locations. Fig. 2 shows an example of landing sequence and trajectory patterns that are expected to be followed by different aircraft depending on their location.

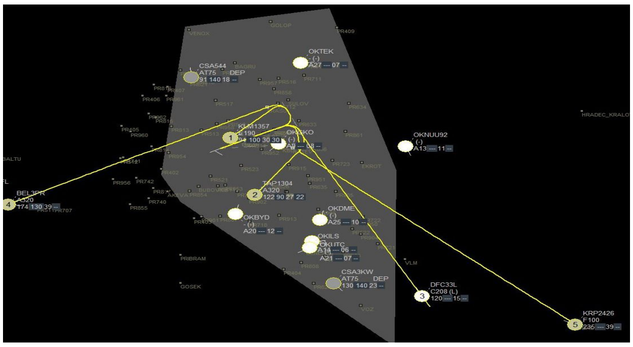


Fig. 2. Expected landing sequences and trajectories for different aircraft approaching Prague airport.

Based on these pre-defined patterns, we designed an "Airport Flight Model", which can predict the future commands to be spoken to a given aircraft based on the **history** (temporal context) of the previously issued commands to that aircraft.

In practice, this model is a Long-Short Term Memory (LSTM) neural network [12, 13] trained on landing sequences of commands, which are reconstructed from data collected in Prague or Vienna airports. That is, we define the input to this model as the timely-ordered sequence of commands, which were issued to a given aircraft since it entered the controlled airspace and until it landed on the runway. The next section shows how this temporal model can be combined with the situational model to generalize the approach proposed in [13].

4. A GENERALIZED CONTEXT-AWARE ASRU SYSTEM

Although the SCR approach (Section 3.2) can significantly improve the performance, it only operates on command values and callsigns. More precisely, if the ASRU hypothesis confuses two commands which take the same attribute but are of different types, the SCR will not be able to correct this misrecognition. e.g. the sequence labeler extracts a "SPEED 220" command instead of a "REDUCE 230", which both take a speed value as attribute. In this case, SCR would be able to correct the command value "220" to "230" but cannot correct the command type "SPEED" to "REDUCE".

This problem can be solved using the TCR approach (Section 3.3). In order to do so, we train this model only on command types without command values, i.e. we only predict the probability of a "REDUCE" command in a given context regardless of the speed value that can be assigned to it. This in fact is a marginalization of the full model (command type+value) on the complete range of values that this command can take. Furthermore, this decision is also justified by the small amount of data available to train the full model, which would result in a vocabulary size of few hundred/thousand, resulting from the rich range of values that each command can take. Building a model only for command types reduces drastically the vocabulary size (40 to 60 different command types).

In order to combine the SCR and TCR models, we consider N-best lists instead of 1-best hypothesis which was used in [8]. Formally, assuming the ASR system produces a list of N hypotheses, let $A = \{A^n\}^N_{n=1} = \{\{A^n_{cs}, \{A^{cs,n}_{com}\}_{com}\}\}^N_{n=1}$ be the set of ATC segments extracted from these hypotheses using sequence labeling (Section 3.2). The combination of SCR and TCR models is done according to

$$\begin{aligned} H &= \underset{n=1,\ldots,N}{\operatorname{argmin}} \{\underset{C\in\mathcal{C}}{\operatorname{argmin}}\{p(A^n, C)\}\} \qquad (2) \\ &= \underset{n=1,\ldots,N}{\operatorname{argmin}} \Big\{ \underset{C\in\mathcal{C}}{\operatorname{argmin}}\{p^s(A^n_{cs}, C_{cs})\} \\ &+ \sum_{A^n_k\in\{A^{cs,n}_{com}\}} \underset{C_j\in\{C^{cs}_{com}\}}{\operatorname{argmin}} \{p^t_{cs}(A^n_k)\cdot p^s(A^n_k, C_j)\}\Big\} \end{aligned}$$

The probability $p(A^n, C)$ combines 1) a situational context based-rescoring probability $p^s(.,.)$, directly derived from the WLD scores used in Section 3.2. This distribution operates on callsigns and command values as explained above, and 2) a temporal context based score $p^t_{cs}(.)$, which estimates the probability distribution over the command type space given the history of issued commands for a callsign cs. In doing so, the situational and temporal context models complement each other, which leads to a generalized model that can successfully rescore callsigns, command types and command values.

5. PRAGUE AND VIENNA DATASETS

The proposed context-based rescoring system is evaluated using recordings of actual ATCOs performing their daily tasks in Prague and Vienna airports. This data was collected as part of the MALORCA[2] project. It consists of 8kHz ATC speech recordings of different noise levels and different radio transmission qualities. In par-

ASRU Systems	Prague Results (Error Rates are in %)					Vienna Results (Error Rates are in %)				
	WER	ConER	CmdER	$\overline{\text{CmdER}}$	R_t(s)	WER	ConER	CmdER	$\overline{\text{CmdER}}$	R_t(s)
SLM (no context)	10.9	17.5	30.9	21.9	1.25	**13.2**	22.3	41.4	30.4	0.90
SLM+Rescoring (N-best=1)	11.2	13.8	19.1	12.8	3.40	17.5	16.4	27.7	20.6	3.22
SLM+Rescoring (N-best=5)	**8.9**	**11.6**	**16.5**	**12.7**	4.65	15.5	**15.5**	**26.3**	**19.8**	3.63
CFG (no context)	18.0	33.1	50.5	37.5	**1.02**	22.1	38.5	58.9	43.1	**0.77**
CFG+Adaptation	17.8	21.9	30.9	23.4	3.57	26.7	29.7	44.1	30.4	1.43
CFG+Rescoring (N-best=1)	19.7	25.3	33.1	25.4	1.87	25.6	27.9	40.1	33.0	1.71
CFG+Rescoring (N-best=10)	19.1	24.4	31.8	24.2	4.57	25.1	26.5	38.5	32.0	2.97

Table 2. ASRU results on 4h of test data from Prague and Vienna airports using different ASRU systems with and without context information.

ticular, the Vienna dataset is very noisy and it can be difficult to understand for humans with no ATC expertise. All commands were issued in English with a mild usage of Czech or Austrian German languages, respectively. In particular for words which do not contain any ATC information such as greetings. Different ATC sessions were recorded over multiple days from each controller. Table 3 presents recording statistics for these two datasets.

The situational context is updated every 5 seconds by the assistant system [10]. Table 3 also reports the context accuracy, i.e. context contains the actual spoken command, and the average context size i.e. number of ATC commands per context file, which can be compared to 239 and 359 used in [7] and [8], respectively.

	Duration (h)		# of Speakers		Context	
	Train	Test	Train	Test	Size	Acc.
Prague	2.1h	1.9h	6	5	650	99.0%
Vienna	5.0h	1.9h	13	6	1600	96.0%

Table 3. Recording statistics for Vienna and Prague datasets including the context accuracy (i.e. contains the actual spoken commands).

6. EXPERIMENTAL SETUP AND ANALYSIS

ASR was performed using the KALDI software [14] and the ASR confidence scores for WLD were generated based on the Minimum Bayesian Risk (MBR) decoding approach [15]. The acoustic model is a DNN/HMM (Deep Neural Network Hidden Markov Model), trained on 150 hours of speech data from the publicly available LIBRISPEECH [16], ICSI [17], AMI [18] and TED-LIUM [19] datasets, which have been extensively used in ASR of conversational speech, and then adapted on Vienna or Prague training data in Table 3. More details about this system can be found in [20]. The SLM is a trigram model trained on a combination of the training data and synthetic data generated from the CFG. The latter defines its rules based on the standard ATC phraseology [11], in addition to most common deviations observed in the training data. The CFG design was guided by the work done in [7, 8].

For evaluation, in addition to conventional WER and Recognition time (R_t), the ATC-specific evaluation metrics *Concept Error Rate* (ConER) and CmdER are used. ConER is restricted to the ATC-relevant semantic concepts of a given utterance, which are extracted using the sequence labeling approach (Section 3.2). A concept can be either a callsign or a command, e.g. AFR2A or REDUCE_250. The CmdER metric requires the entire sequence of concepts to be correct. In the case where the sequence labeling system fails in extracting ATC segments, it returns NO_CALLSIGN or NO_COMMAND, which are counted as misrecognition, even though they have no impact on the planning system (no information). Therefore, we also report the CmdER after excluding these utterances (noted $\overline{\text{CmdER}}$) to estimate the misrecognition rate which negatively affects the planning system.

Table 2 reports the ASRU results for Vienna and Prague test data with and without context information. The approach "CFG+Adaptation" is the one proposed in [7]. Furthermore, using an N-best=1 is equivalent to the system proposed in [8], which does not use temporal context. In this case, the recognized ATC segment contains (at most) one command type. Thus, the TCR is not used.

The results clearly confirm the conclusions reported in [8]. That is, SLM clearly outperforms the CFG-based system with and without context information. This observation highlights the importance of the probability distribution learned by SLM but ignored by CFG, which uses a uniform distribution over words and commands. Moreover, SLM automatically captures deviations from standard phraseology present in the data, whereas CFG requires a manual addition.

We can also conclude from these results that context information strongly improves the ATC-related metrics (ConER, CmdER and $\overline{\text{CmdER}}$), whereas it slightly improves or worsens the WER of either system. This is an expected outcome given that the proposed approach is mainly designed to improve the ConER (and therefore also the CmdER), by directly extracting and correcting ATC segments from the recognized hypotheses. Correcting such segments, however, does not necessarily mean improving the word-level recognition. This is particularly true in cases where the controller deviates from standard phraseology [11], which was used to build the context-to-word mapping (Section 3.2), e.g. dropping the word "decimal" while issuing the frequency 133.2 ="one three three decimal two". These cases were very common, particularly in Vienna data. Furthermore, increasing the N-best list size leads to further improvements for all systems. This observation highlights the advantages of the proposed generalized system compared to the one proposed in [8] (N-best=1). In fact, testing the TCR component alone leads to an accuracy (prediction of the command type) of 59% for Prague and 55% for Vienna, with a mean rank of 2.4 and 2.7, respectively.

These experiments also show that data and context quality are very crucial. More particularly, the Prague speech data is less noisy compared to Vienna data and largely benefits from the smaller and more accurate situational context (Table 3). Moreover, comparing CmdER and $\overline{\text{CmdER}}$ shows an average degradation of $\approx 10\%$. This reflects the need for a better sequence labeler to extract the ATC segments. The recognition time R_t, however, is within a real-time range given that ATC utterances are $\approx 3.7s$ long on average.

7. CONCLUSIONS AND FUTURE WORK

We proposed a context-aware ASRU system for ATC domain, which combines situational context acquired through an ATC assistance system, and temporal context given by the history of issued commands. Experiments conducted on real data from Prague and Vienna airports showed a significant reduction of the command error rate. Our future work will focus on investigating different sequence labeling approaches, which seem to be a cornerstone for improving the performance of the overall system.

8. REFERENCES

[1] Geert-Jan M. Kruijff, Pierre Lison, Trevor Benjamin, Henrik Jacobsson, Hendrik Zender, and Ivana Kruijff-Korbayová, "Situated dialogue processing for human-robot interaction," in *Cognitive Systems*, vol. 8 of *Cognitive Systems Monographs*, chapter 8, pp. 311–364. Springer Verlag, Berlin/Heidelberg, Germany, 2010.

[2] Sheryl R. Young, Wayne H. Ward, and Alexander G. Hauptmann, "Layering predictions: Flexible use of dialog expectation in speech recognition," in *Proceedings of the 11th International Joint Conference on Artificial Intelligence. Detroit, MI, USA, August 1989*, 1989, pp. 1543–1549.

[3] S. R. Young, A. G. Hauptmann, W. H. Ward, E. T. Smith, and P. Werner, "High level knowledge sources in usable speech recognition systems," *Commun. ACM*, vol. 32, no. 2, pp. 183–194, Feb. 1989.

[4] Christian Fügen, Hartwig Holzapfel, and Alex Waibel, "Tight coupling of speech recognition and dialog management - dialog-context dependent grammar weighting for speech recognition," in *INTERSPEECH 2004 - ICSLP, 8th International Conference on Spoken Language Processing, Jeju Island, Korea, October 4-8, 2004*.

[5] Katherine Everitt, Susumu Harada, Jeff A. Bilmes, and James A. Landay, "Disambiguating speech commands using physical context," in *Proceedings of the 9th International Conference on Multimodal Interfaces, ICMI 2007, Nagoya, Aichi, Japan, November 12-15, 2007*, 2007, pp. 247–254.

[6] Todd Shore, Friedrich Faubel, Hartmut Helmke, and Dietrich Klakow, "Knowledge-based word lattice rescoring in a dynamic context," in *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012*, 2012, pp. 1083–1086.

[7] Anna Schmidt, Youssef Oualil, Oliver Ohneiser, Matthias Kleinert, Marc Schulder, Arif Khan, and Hartmut Helmke, "Context-based recognition network adaptation for improving on-line asr in air traffic control," in *2014 IEEE Spoken Language Technology Workshop (SLT 2014)*, 2014, pp. 2–6.

[8] Youssef Oualil, Marc Schulder, Hartmut Helmke, Anna Schmidt, and Dietrich Klakow, "Real-time integration of dynamic context information for improving automatic speech recognition," in *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*, 2015, pp. 2107–2111.

[9] Hartmut Helmke, Youssef Oualil, Jürgen Rataj, Thorsten Mühlhausen, Oliver Ohneiser, Heiko Ehr, Matthias Kleinert, and Marc Schulder, "Assistant-based speech recognition for ATM applications," in *Proceedings of* 11^{th} *USA/Europe ATM R&D Seminar (ATM2015)*, Lisbon, Portugal, June 2015.

[10] Hartmut Helmke, Ronny Hann, Maria Uebbing-Rumke, Dennis Müller, and Dennis Wittkowski, "Time-based arrival management for dual threshold operation and continuous descent approaches," in *Proceedings of* 8^{th} *USA/Europe ATM R&D Seminar (ATM2009)*, Napa, California, USA, June - July 2009.

[11] "All clear phraseology manual," in *Eurocontrol, Brussels, Belgium*, April 2011.

[12] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney, "LSTM neural networks for language modeling," in *13th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, Portland, OR, USA, Sep. 2012, pp. 194–197.

[13] Y. Oualil and D. Klakow, "A neural network approach for mixing language models," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 5710–5714.

[14] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The Kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. Dec. 2011, IEEE Signal Processing Society.

[15] Vaibhava Goel and William J. Byrne, "Minimum bayes-risk automatic speech recognition," *Computer Speech & Language*, vol. 14, no. 2, pp. 115–135, 2000.

[16] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[17] Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, and Chuck Wooters, "The ICSI meeting corpus," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2003.

[18] Jean Carletta, "Announcing the AMI meeting corpus," *The ELRA Newsletter*, vol. 11, no. 1, pp. 3–5, 2006.

[19] Anthony Rousseau, Paul Deléglise, and Yannick Estève, "Enhancing the TED-LIUM Corpus with Selected Data for Language Modeling and More TED Talks," in *Proc. of the 9th International Conference on Language Resources and Evaluation (LREC)*, 2014, pp. 3935–3939.

[20] Ajay Srinivasamurthy, Petr Motlicek, Ivan Himawan, Gyorgy Szaszak, Youssef Oualil, and Hartmut Helmke, "Semi-supervised learning with semantic knowledge extraction for improved speech recognition in air traffic control," in *INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association*, Aug. 2017.

5.9 Paper 9: [Koc+21b]

Boosting of contextual information in ASR for air-traffic call-sign recognition

Martin Kocour[1], *Karel Veselý*[1], *Alexander Blatt*[2], *Juan Zuluaga Gomez*[3,4], *Igor Szöke*[1], *Jan "Honza" Černocký*[1], *Dietrich Klakow*[2]*and Petr Motlíček*[3]

[1]Brno University of Technology, Faculty of Information Technology, Speech@FIT, Czechia
[2]Saarland University, Saarbrücken, Germany
[3]Idiap Research Institute, Martigny, Switzerland
[4]Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland

ikocour@fit.vutbr.cz, iveselyk@fit.vutbr.cz

Abstract

Contextual adaptation of ASR can be very beneficial for multi-accent and often noisy Air-Traffic Control (ATC) speech. Our focus is call-sign recognition, which can be used to track conversations of ATC operators with individual airplanes. We developed a two-stage boosting strategy, consisting of HCLG boosting and Lattice boosting. Both are implemented as WFST compositions and the contextual information is specific to each utterance. In HCLG boosting we give score discounts to individual words, while in Lattice boosting the score discounts are given to word sequences. The context data have origin in surveillance database of OpenSky Network. From this, we obtain lists of call-signs that are made more likely to appear in the best hypothesis of ASR. This also improves the accuracy of the NLU module that recognizes the call-signs from the best hypothesis of ASR.

As part of ATCO^2 project, we collected liveatc_test_set2. The boosting of call-signs leads to 4.7% absolute WER improvement and 27.1% absolute increase of Call-Sign recognition Accuracy (CSA). Our best result of 82.9% CSA is quite good, given that the data is noisy, and WER 28.4% is relatively high. We believe there is still room for improvement.

Index terms: Air Traffic Control, Automatic Speech Recognition, Contextual Adaptation, Call-sign Recognition, Call-sign Detection, OpenSky Network.

1. Introduction

The purpose of aviation call-signs is to identify airplanes in Air Traffic Control (ATC) procedures. Many ATC messages are currently conveyed by voice over noisy VHF channel. If we had perfect call-sign recognition, we could easily track conversations of *pilots* with *ATC operators* in the shared audio channel. The tracking would be useful for post-analysis of recordings, or possibly for real-time ATC systems of the airports.

Recently, call-sign detection was an evaluation task in *Airbus Air Traffic Control challenge* [1, 2]. We redefined the task from detection to call-sign *recognition*, as we *recognize* the ICAO call-sign codes (e.g. `TVS123AB`) from the best ASR hypothesis. Then, the call-sign code can be directly interfaced to radar or other system. From the perspective of our paper, the call-sign recognition module is a black-box, and we focus on improving ATC-ASR (i.e. ASR for ATC data) by leveraging contextual information. The context we use are call-sign lists for given location and time, and these lists are queried from OpenSky Network (OSN) database [3, 4].

Several works are addressing the use of contextual information for ATC-ASR [5, 6, 7]. Shore et al. [5] introduced a lattice-rescoring mechanism, penalizing call-signs not present in the current radar situation. Schmidt et al. [6] built a grammar-based ASR, in which the search-space is limited to all command-predictions for actual radar situation. The context adaptation is continuous and integrated with on-line ASR. Later, in [7] Oualil et al. compare grammar-based and n-gram-based language modelling in ASR, showing n-grams as better. The n-grams cover well the irregularities of real ATC speech. Again, the context adaptation is continuous, and Weighted Levenshtein Distance algorithm is used to select command prediction closest to the ASR output. These works inspired us to focus on continuous adaptation and its integration into ATC-ASR with n-gram language models.

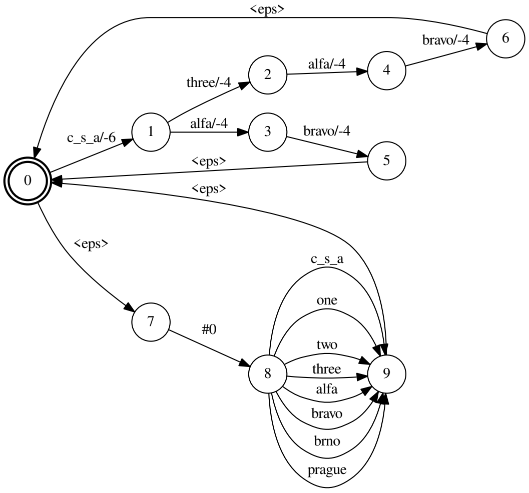


Figure 1: *Topology of WFST graph for boosting of lattices.*

Otherwise, a significant inspiration for our Lattice boosting was the work on *composition-based on-the-fly rescoring* [8], where rapid rescoring is done on unpruned pseudo-deterministic word-lattices. LM weights are adjusted for a small set of n-grams representing contextual information. Later, in *rescoring-aware beam search* [9], a secondary larger beam was introduced into the decoder generating lattices. The secondary beam is applied to the context represented as n-grams that are later biased by rescoring. The purpose is to reduce a chance that the context is pruned-out in the lattice generation. With the very same motivation, we introduce our on-the-fly HCLG graph boosting. Here, the score discounts are given to single words relevant to the context. Our technique is simpler to implement.

2. Call-sign boosting in ASR system

As *call-sign recognition* has many practical use-cases for processing ATC data, we focus on improving *Call-Sign recognition Accuracy* (CSA). We improve CSA by targeted boosting of certain words, or word-strings. We give them score discounts into language model scores, which is done by means of WFST composition [8]. The boosted expressions are thus made more likely to appear in the best hypothesis of ASR. This approach is natural for Weighted Finite State Transducer (WFST) based ASR systems. And Kaldi [10] ASR systems do use OpenFst [11] for representing WFSTs.

The composition is done with a boosting graph that holds score discounts. The original language model log-scores are still used in the decoding process, as the score discounts are added as their offsets.

The boosting graph is distinct for each utterance, so we have to be aware that composition can easily become a computationally demanding operation. The complexity of WFST composition depends on numbers of states in the two operands, the number of outgoing arcs from states and a degree of non-determinism[1].

2.1. Obtaining the call-sign lists

For boosting, we need lists of candidate call-signs, which are capturing the short-term traffic situation. These can be obtained in a dynamic way from a radar system, or in a static way from a historical database of traffic monitoring. Our partner in the ATCO^2 project - OpenSky Network - provides an access to its database of surveillance data [3]. The surveillance data are collected from ADS-B receivers operated by a network of volunteers. The queries for call-sign lists are bounded both spatially and with a time-frame [12].

For evaluation sets from HAAWAII project, we use call-sign lists from radar system of the airport.

2.2. Verbalizing a call-sign

An example of the original ICAO call-sign code format from the lists is: `TVS123AB`. This can be verbalized in several ways. Our verbalization is an extension of ICAO standard [13]:

```
skytravel one two three alfa bravo
skytravel three alfa bravo
skytravel alfa bravo
skytravel one alfa bravo
skytravel one two bravo
tango victor sierra one two three alfa bravo
one two three alfa bravo
three alfa bravo
alfa bravo
```

The translation `TVS -> skytravel` is done according to a look-up table of airline designators. The rest of the code should be read as isolated numbers, and the suffix of letters is 'spelled' with ICAO alphabet. Shortening right after the airline designator is possible. Spelling of `TVS` with ICAO alphabet is also acceptable in the standard. Some common non-standard variations include shortening the airline designator `lufthansa -> hansa`, or omitting it if the situation is not ambiguous. We support also other non-standard call-sign shortenings, and number expansions of type `777 -> triple seven`. Airplanes not serving in airlines have registration number as a call-sign. The registration has a prefix that encodes country, which is spelled by ICAO alphabet (e.g. `OK` for Czech Republic, or `HB` for Switzerland).

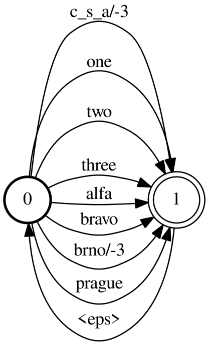


Figure 2: *Topology of WFST graph for boosting the recognition network HCLG.*

2.3. Lattice boosting

The *lattice boosting* is done as composition:

$$L' = L \circ B \;, \tag{1}$$

where L is the original lattice, and B is boosting graph. The boosting graph B is specific for each utterance.

The toy-example boosting graph in Figure 1 has a lower part with all the words in a lexicon on parallel arcs. This ensures no word sequence being dropped from the original lattice by the composition. There is also a *phi* input symbol `#0` on the 'entrance' arc to the lower part. The upper part encodes word sequences of call-signs, the score discounts -4 or -6 are on word links. We intended to experiment with a combination of per-word and per call-sign score discounts. The phi symbol allows entering lower part only if the sub-path cannot be matched by the upper part of boosting graph.

The composition is run in batch mode for whole test-set, but it could be also done on-line after finalizing the lattice. The composition is fast because both the lattices and boosting graphs are relatively small.

The word sequence must be present in the lattice in order to be boosted. The Lattice Oracle WER is computed from lattice-path that is closest to the correct transcript, and the oracle alignments can hint us of problematic words.

2.4. HCLG boosting

The *HCLG boosting* is done as composition:

$$HCLG' = HCLG \circ B \;, \tag{2}$$

where $HCLG$ [2] is the pre-compiled recognition network, and B is another type of boosting graph. The $HCLG'$ graph is used for lattice-generation, and the boosting graph B is again specific for each utterance. The composition of B with $HCLG$ graph is done on-the-fly immediately before initializing the decoder.

The toy-example boosting graph in Figure 2 boosts individual words. In the figure it is `c_s_a` and `brno` which get the score discount -3, other words have no discount. Also, note the `<eps>` back-link into state 0.

[1] `http://www.openfst.org/twiki/bin/view/FST/ComposeDoc`

[2] HCLG is composed from 'H' with HMM topology, 'C' for context dependency, 'L' with lexicon and 'G' for language model (grammar), more info in `https://kaldi-asr.org/doc/graph.html`

database	**hours**	**accents**	**ref.**
AIRBUS	38.9	French	[14]
HIWIRE	28.7	French, Greek, Italian and Spanish	[15]
LDC ATCC	26.2	American English	[16]
MALORCA	7.9	Austrian German	[17, 18]
N4 NATO	10.7	Canadian, German, Dutch, British	[19]
ATCOSIM	10.7	German, Swiss German + French	[20]
UWB ATCC	13.2	Czech	[21, 22]
Total sum:	136.3		

Table 1: *Audio databases for training the ASR models.*

test-set	**hours**	**description**
airbus_dev	1.03	custom held-out set from Airbus challenge data, mostly from 'lfbo' airport, both operator and pilot speech
malorca_vienna	1.93	test-set from project MALORCA, Vienna airport 'loww', no pilot speech
liveatc_test_set2	0.88	our own collection and manual transcription of LiveATC data, mostly Zurich airport 'lszh' plus some 'eidw' and 'katl', contains operator and pilot speech, some parts are noisy
haawaii_bikf haawaii_egll	5.31 6.85	data from HAAWAII project, Keflavik airport 'bikf' and London Heathrow 'egll', both operator and pilot speech, 'egll' has more noise

Table 2: *Audio data for testing ASR and Call-sign recognition.*

The purpose of HCLG boosting is to decrease the Lattice Oracle WER, so that the recall of call-signs in Lattice boosting increases. And, by boosting more call-signs in lattices, the final WER improves as well.

In the HCLG graph, we cannot boost word-strings as in case of using graph from Figure 1. The composition would be prohibitively slow, about 5 minutes per composition. By simplifying the boosting graph to topology from Figure 2, we already got an affordable increase of processing time of 20-30% on top of lattice-generation time.

We also apply epsilon-removal on B, prior to the composition, which reduces the composition run-time. In fact B could be a single state WFST right from the beginning, the second state is added for easier visualisation.

We believe that only rare, context-specific, individual words should be boosted in HCLG boosting. As we focus on call-sign recognition, we are boosting only the airline designator codewords like `skytravel`, `c_s_a` or `air_berlin`.

An alternative strategy to HCLG boosting would be to boost the `G.fst` and do the on-the-fly composition with `HCL.fst` graph as is done in Kaldi tool `nnet3-latgen-faster-lookahead`. The cascade of on-the-fly compositions $HCL \circ (G \circ B)$ would introduce some latency too. We will consider exploring this possibility as a follow-up work.

Training data	liveatc_test_set2	airbus_dev	malorca_vienna
with malorca	33.1	8.3	**4.7**
w/o malorca	35.1	8.4	**8.9**

Table 3: *Simulating deployment of ASR to 'malorca_vienna' as a 'new' airport, WER% results.*

Beams	WER%			Lattice Oracle	
	baseline	lattice boost	HCLG+lat boost	baseline	HCLG boost
b=10, lb=5	32.9	31.2	30.2	*21.4*	*19.8*
b=15, lb=8	33.1	30.0	28.8	15.2	13.9
b=20, lb=11	33.0	29.1	**28.4**	12.0	11.1
b=25, lb=13	33.1	**28.9**	**28.4**	*11.3*	*10.7*

Table 4: *Effect of tuning the beam-width for 'lattice boosting' and 'HCLG + lattice boosting', data-set liveatc_test_set2.*

3. Experimental setup

3.1. Audio databases

For *training* the ASR, we pre-processed 7 audio databases of English ATC audio data, see Table 1. Various accents are present. Some data are from simulated scenarios (HIWIRE, N4 NATO, ATCOSIM), while other audio is from real traffic. Particularly the unification of transcripts ended up being a challenging task.

For *testing*, we use 5 different sets, see Table 2. The test-sets differ in quality of signal : 'airbus_dev', 'malorca_vienna' and 'haawaii_bikf' are clean, 'liveatc_test_set2' is quite noisy and 'haawaii_egll' contains some moderate noise. Next, 'malorca_vienna' contains no pilot speech. And further, the airports from 'liveatc_test_set2', 'haawaii_bikf' and 'haawaii_egll' are not present in training data of our ASR system.

Even though the ATC messages should follow a standard [13], we had to normalize the transcripts as follows: a) to use same ICAO alphabet, b) to use only one variant of word-splits in common expressions (e.g. 'take off' 'take-off' → 'takeoff', 'flightlevel' → 'flight level', etc.), c) to standardize the airline designators according to a "correct" table and map spaces and dashes to underscores (e.g. 'norshuttle' 'nor shuttle' → 'nor_shuttle', or 'fly niki' 'fly_niki' 'fly-niki' → 'flyniki').

3.2. Baseline ASR system

We use a 'hybrid' speech-to-text recognizer based on Kaldi [10] trained with Lattice-free MMI [23]. The neural network has 6 'conv-relu-batchnorm-layer' convolutional layers followed by 9 'tdnnf-layer' semi-orthogonal components [24]. As usual, there are two pre-final layers and two output layers: one for LF-MMI objective, the second for frame cross-entropy objective. In total, the model has 12.93 million trainable parameters, and the number of left biphone tied-states is 1680. The input features are high-resolution Mel-frequency cepstral coefficients (MFCC) with online Cepstral mean normalization (CMN). The features are extended with online i-vectors [25, 26].

Lexicon: The positive side of ATC-ASR is that the vocabulary is relatively small compared to general purpose ASR. In our case, there are 28.4k unique tokens in lexicon, out of that 15.3k are 5-letter waypoints, and 5.2k are airline designators for call-signs. We tried to create a rich vocabulary in advance to minimize the OOV problem.

We used Phonetisaurus [27] to build a grapheme to phoneme model from Librispeech lexicon [28]. We limited the vocabulary to ATC word-list gathered from 7 training databases, our test-sets, and some other pre-collected word-lists (airline

	Baseline		Lattice boost.		HCLG+Lat. boost.		Oracle
	CSA	WER	*CSA*	WER	*CSA*	WER	*CSA*
liveatc_test_set2	*53.5*	33.1	*75.6*	28.9	*80.6*	28.4	*90.0*
malorca_vienna	*84.4*	8.9	*86.5*	8.1	*88.1*	7.5	*90.5*
haawaii_bikf	-	30.6	-	29.4	-	28.9	-
haawaii_egll	-	20.8	-	19.3	-	18.8	-

Table 5: *Call-Sign recognition Accuracy % (CSA) and Word Error Rate % (WER) for 4 test sets and 2 types of ASR boosting: 'Lattice boosting' and 'HCLG + Lattice boosting'. The Oracle CSA is calls-sign recognition from ground truth transcripts.*

designators, waypoints, airports, cities, countries, etc.).

The table of airline designators was prepared from Wikipedia page[3]. We cross-checked some items with other public databases. Recently, we found an FAA document[4], which could be used in future. The list of European waypoints was obtained from *traffic* [29] python project.

Language model: We use 3-gram language model built by interpolating several LMs with SRI-LM [30]. The mixing coefficients are tuned on entire 'liveatc_test_set1' (i.e. a set different from liveatc_test_set2) complemented with a fragment of 'airbus_dev' and 'malorca_vienna' test-sets. We build one LM from each training corpus transcripts (except HIWIRE and N4 NATO whose transcripts have limited variability).

An additional LM for interpolation is built from 'extra_data', i.e. a collection of: a) expanded call-signs from OSN database with 2019 world-wide traffic[5], b) all possible runway number combinations, c) European waypoints in typical idioms, and d) pre-collected word-lists previously added to lexicon.

4. Results

4.1. Deploying ASR to new airport, simulation

An ideal ATC-ASR system should generalize to a 'new' airport. In practice, the training data come from some airport, and performance for that airport is better than for some 'new' airport. We quantified this effect in Table 3.

By excluding malorca data from the training (acoustic model, language model and lexicon), the WER nearly doubles 4.7 → 8.9 for malorca_vienna test set. For other test sets, the error rate almost did not change. The malorca data consist of purely ATC operator speech, and including pilot speech would further increase the WER. Our boosting experiments are done with an ASR system that had malorca data excluded, to simulate the 'new' airport scenario.

4.2. Call-sign boosting, ASR performance, tuning beams

Next, we experiment with call-sign boosting. The call-sign words represent roughly 25% of reference transcript text. We evaluate Lattice boosting and a cascade of HCLG boosting and Lattice boosting. The liveatc_test_set2 is used to tune the beam widths and values of score discounts.

Table 4 shows a significant improvement 4.7% of WER (33.1 → 28.4) from the combination of HCLG boosting and Lattice boosting. If we do only Lattice boosting, the performance gain is little smaller (4.2%). Further widening the beams can, to some extent, compensate for not doing the HCLG boosting, but the lattices also grow larger. With 'lb=8' the liveatc_test_set2 lattices have 12MB, with 'lb=11' 89MB, and for 'lb=13' 192MB.

In first column, 'b=' stands for `--beam` and 'lb=' for `--lattice-beam`. The default values from Kaldi are 'b=15, lb=8'. Larger beams lead to better performance, but the system becomes slower. We also see the effect of HCLG boosting of airline code-words on Lattice Oracle WER. The improvements are ranging from 1.6% absolute for smaller lattices generated with narrow beams to 0.6% for wide beams.

4.3. Callsign accuracy performance

The ASR output is processed by call-sign recognition module, which is an End2End neural network that translates text directly into ICAO call-sign code like `TVS123AB`. The performance is measured as Call-Sign recognition Accuracy (CSA). The call-sign recognizer uses list of candidate call-signs as contextual information, while it still can synthesize a new call-sign not present in the list.

In Table 5, we see that WER improvements consistently translate into CSA improvements. On liveatc_test_set2 we have a huge improvement from 53.5 to 80.6. For malorca_vienna the absolute CSA improvement is smaller, nevertheless the gain from 84.4 to 88.1 removed 60.7% of the gap spanning from baseline to oracle CSA. For test-sets from HAAWAII project, we have only WER scores that show consistent improvements. For evaluation of call-sign recognition, we kept only utterances where the true call-sign was present also in the traffic monitoring data. This reduces the risk of having a wrong call-sign in the ground-truth annotation.

5. Conclusions

Inspired by other works on contextual adaptation of WFST-based ASR systems, we applied a cascade of on-the-fly HCLG boosting of individual words and Lattice boosting of word sequences. The boosted elements appear more likely as part of the best ASR hypothesis.

We focused on call-sign recognition from air-traffic control speech. Our boosting improved dramatically both the Word Error Rate and Call-sign recognition accuracy, especially for noisy test-set like liveatc_test_set2 : WER -4.7% absolute, Call-sign accuracy +27.1% absolute in Table 5. The proposed technique of giving score discounts to certain words or word sequences in ASR inference is generic and can be used in other domains.

In future, we plan to extend contextual adaptation to more types of content, for example waypoints, geographical names, or frequent expressions in local language.

6. Acknowledgements

We would like to thank to Hartmut Helmke for early feedback and DLR for providing annotated audio data from HAAWAII project. The work was supported by European Union's Horizon 2020 projects No. 864702 - ATCO2 (all auth.) and No. 884287 HAAWAII (BUT and IDIAP). Part of high-performance computation run on IT4I supercomputer and was supported by the Ministry of Education, Youth and Sports of the Czech Republic through e-INFRA CZ (ID:90140).

[3] `https://en.wikipedia.org/wiki/List_of_airline_codes`

[4] `https://www.faa.gov/documentLibrary/media/Order/7340.2J_Chg_1_dtd_10_10_19.pdf`

[5] `https://zenodo.org/record/3901482#.X5cK9k_0m_4`

7. References

[1] T. Pellegrini, J. Farinas, E. Delpech, and F. Lancelot, "The Airbus Air Traffic Control Speech Recognition 2018 Challenge: Towards ATC Automatic Transcription and Call Sign Detection," in *Interspeech 2019, Graz, Austria, September 2019*. ISCA, 2019, pp. 2993–2997. [Online]. Available: https://doi.org/10.21437/Interspeech.2019-1962

[2] V. Gupta, L. Rebout, G. Boulianne, P. A. Ménard, and J. Alam, "CRIM's Speech Transcription and Call Sign Detection System for the ATC Airbus Challenge Task," in *Interspeech 2019, Graz, Austria, September 2019*. ISCA, 2019, pp. 3018–3022. [Online]. Available: https://doi.org/10.21437/Interspeech.2019-1131

[3] J. Sun and J. M. Hoekstra, "Integrating pyModeS and OpenSky Historical Database," in *Proceedings of the 7th OpenSky Workshop*, vol. 67, 2019, pp. 63–72.

[4] M. Schäfer, M. Strohmeier, V. Lenders, I. Martinovic, and M. Wilhelm, "Bringing up OpenSky: A large-scale ADS-B sensor network for research," in *Proceedings of the 13th IEEE/ACM International Symposium on Information Processing in Sensor Networks*. IEEE Press, 2014, pp. 83–94.

[5] T. Shore, F. Faubel, H. Helmke, and D. Klakow, "Knowledge-based word lattice rescoring in a dynamic context," in *INTERSPEECH 2012, Portland, Oregon, USA, September 2012*. ISCA, 2012, pp. 1083–1086. [Online]. Available: http://www.isca-speech.org/archive/interspeech_2012/i12_1083.html

[6] A. Schmidt, Y. Oualil, O. Ohneiser, M. Kleinert, M. Schulder, A. Khan, H. Helmke, and D. Klakow, "Context-based recognition network adaptation for improving on-line ASR in Air Traffic Control," in *2014 IEEE SLT 2014, South Lake Tahoe, NV, USA, December 2014*. IEEE, 2014, pp. 13–18. [Online]. Available: https://doi.org/10.1109/SLT.2014.7078542

[7] Y. Oualil, M. Schulder, H. Helmke, A. Schmidt, and D. Klakow, "Real-time integration of dynamic context information for improving automatic speech recognition," in *INTERSPEECH 2015, Dresden, Germany, September 2015*. ISCA, 2015, pp. 2107–2111. [Online]. Available: http://www.isca-speech.org/archive/interspeech_2015/i15_2107.html

[8] K. B. Hall, E. Cho, C. Allauzen, F. Beaufays, N. Coccaro, K. Nakajima, M. Riley, B. Roark, D. Rybach, and L. Zhang, "Composition-based on-the-fly rescoring for salient n-gram biasing," in *INTERSPEECH 2015, Dresden, Germany, September 2015*. ISCA, 2015, pp. 1418–1422. [Online]. Available: http://www.isca-speech.org/archive/interspeech_2015/i15_1418.html

[9] I. Williams and P. S. Aleksic, "Rescoring-aware beam search for reduced search errors in contextual automatic speech recognition," in *INTERSPEECH 2017, Stockholm, Sweden, August 2017*. ISCA, 2017, pp. 508–512. [Online]. Available: http://www.isca-speech.org/archive/Interspeech_2017/abstracts/1671.html

[10] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi Speech Recognition Toolkit," in *IEEE ASRU 2011*, Dec.

[11] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri, "OpenFst: A general and efficient weighted finite-state transducer library," in *Implementation and Application of Automata, 12th International Conference, CIAA 2007, Prague, Czech Republic*.

[12] J. Zuluaga-Gomez, K. Veselý, A. Blatt *et al.*, "Automatic call sign detection: Matching air surveillance data with air traffic spoken communications," in *Proceedings of the 8th OpenSky Symposium 2020*, vol. 2020, no. 59. MDPI, 2020, pp. 1–10.

[13] "Aeronautical Telecommunications, Annex 10, Volume II," ser. Edition 6. International Civil Aviation Organization (ICAO), october 2001. [Online]. Available: https://www.icao.int/Meetings/anconf12/Document%20Archive/AN10_V2_cons%5B1%5D.pdf

[14] E. Delpech, M. Laignelet, C. Pimm, C. Raynal, M. Trzos, A. Arnold, and D. Pronto, "A real-life, French-accented corpus of air traffic control communications," in *Proceedings LREC 2018*.

[15] J. Segura, T. Ehrette, A. Potamianos, D. Fohr, I. Illina, P. Breton, V. Clot, R. Gemello, M. Matassoni, and P. Maragos, "The HIWIRE database, a noisy and non-native English speech corpus for cockpit communication," *Online. http://www. hiwire. org*, 2007.

[16] J. Godfrey, "The Air Traffic Control Corpus (ATC0) - LDC94S14A," 1994. [Online]. Available: https://catalog.ldc.upenn.edu/LDC94S14A

[17] M. Kleinert, H. Helmke, G. Siol, H. Ehr, A. Cerna, C. Kern, D. Klakow, P. Motlicek, Y. Oualil, M. Singh *et al.*, "Semi-supervised adaptation of assistant based speech recognition models for different approach areas," in *37th Digital Avionics Systems Conference (DASC)*. IEEE, 2018, pp. 1–10.

[18] A. Srinivasamurthy, P. Motlicek, I. Himawan, G. Szaszak, Y. Oualil, and H. Helmke, "Semi-supervised learning with semantic knowledge extraction for improved speech recognition in air traffic control," in *Proc. INTERSPEECH 2017*.

[19] C. Swail, L. Benarousse, E. Geoffrois, J. Grieco, R. Series, H. Steeneken, H. Stumpf, and D. Thiel, "The NATO Native and Non-Native (N4) Speech Corpus," 01 2003.

[20] K. Hofbauer, S. Petrik, and H. Hering, "The ATCOSIM corpus of non-prompted clean air traffic control speech." in *LREC*, 2008.

[21] L. Šmídl, J. Švec, D. Tihelka, J. Matoušek, J. Romportl, and P. Ircing, "Design and development of speech corpora for air traffic control training," in *Proceedings LREC 2018, Miyazaki, Japan, May 2018*. (ELRA), 2018. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2018/summaries/41.html

[22] L. Šmídl, J. Švec, D. Tihelka, J. Matoušek, J. Romportl, and P. Ircing, "Air traffic control communication (ATCC) speech corpora and their use for ASR and TTS development," *Language Resources and Evaluation*, vol. 53, no. 3, pp. 449–464, 2019.

[23] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in *INTERSPEECH 2016, San Francisco, CA, USA, September 2016*. ISCA, 2016, pp. 2751–2755. [Online]. Available: https://doi.org/10.21437/Interspeech.2016-595

[24] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks," in *Proceedings of INTERSPEECH 2018*, 09 2018, pp. 3743–3747.

[25] V. Peddinti, G. Chen, V. Manohar, T. Ko, D. Povey, and S. Khudanpur, "JHU ASpIRE system: Robust LVCSR with TDNNS, iVector adaptation and RNN-LMS," in *2015 IEEE ASRU 2015, Scottsdale, AZ, USA, December 2015*. IEEE, 2015, pp. 539–546. [Online]. Available: https://doi.org/10.1109/ASRU.2015.7404842

[26] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *2013 IEEE ASRU, Olomouc, Czech Republic, December 2013*. IEEE, 2013, pp. 55–59. [Online]. Available: https://doi.org/10.1109/ASRU.2013.6707705

[27] J. R. Novak, N. Minematsu, and K. Hirose, "Phonetisaurus: Exploring grapheme-to-phoneme conversion with joint n-gram models in the WFST framework," *Natural Language Engineering*, vol. 22, no. 6, pp. 907–938, 2016.

[28] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[29] X. Olive and L. Basora, "A Python Toolbox for Processing Air Traffic Data: A Use Case with Trajectory Clustering," in *Proceedings of the 7th OpenSky Workshop 2019, Zurich, Switzerland, November 21-22, 2019*, ser. EPiC Series in Computing, vol. 67, 2019, pp. 73–84.

[30] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *ICSLP2002 - INTERSPEECH 2002, Denver, Colorado, USA, September 16-20, 2002*, J. H. L. Hansen and B. L. Pellom, Eds. ISCA, 2002. [Online]. Available: http://www.isca-speech.org/archive/icslp_2002/i02_0901.html

5.10 Paper 10: [Lec+12]

Supervised and unsupervised Web-based language model domain adaptation

Gwénolé Lecorvé[1], *John Dines*[1,2], *Thomas Hain*[3], *Petr Motlicek*[1]

[1]Idiap Research Institute, Martigny, Switzerland
[2]Koemei, Martigny, Switzerland
[3]University of Sheffield, Sheffield, United Kingdom

glecorve@idiap.ch, dines@idiap.ch, t.hain@dcs.shef.ac.uk, motlicek@idiap.ch

Abstract

Domain language model adaptation consists in re-estimating probabilities of a baseline LM in order to better match the specifics of a given broad topic of interest. To do so, a common strategy is to retrieve adaptation texts from the Web based on a given domain-representative seed text. In this paper, we study how the selection of this seed text influences the adaptation process and the performances of resulting adapted language models in automatic speech recognition. More precisely, the goal of this original study is to analyze the differences of our Web-based adaptation approach between the supervised case, in which the seed text is manually generated, and the unsupervised case, where the seed text is given by an automatic transcript. Experiments were carried out on data sourced from a real-world use case, more specifically, videos produced for a university YouTube channel. Results show that our approach is quite robust since the unsupervised adaptation provides similar performance to the supervised case in terms of the overall perplexity and word error rate.

Index Terms: Language model, domain adaptation, supervision, Web data

1. Introduction

The n-gram language model (LM) of most automatic speech recognition (ASR) systems is usually trained on a large multi-topic text collection. As a consequence, this LM is not optimal to transcribe spoken documents dealing with a given specific domain. To solve this problem, domain LM adaptation seeks to re-estimate the n-gram probabilities of the baseline LM in order to fit the specifics of the considered domain. The ultimate goal of this adaptation is to improve the quality of ASR transcripts.

Nowadays, a standard approach for LM domain adaptation consists of using the Web as an open corpus in order to retrieve domain-specific data providing accurate statistics for n-gram re-estimation [1, 2, 3, 4, 5]. The process of the Web-based adaptation can be split into the following steps: first, one has to extract queries from a given text that is representative of the domain of interest—this text is called the *seed text* ; then Web pages are retrieved by submitting the queries to a Web search engine ; finally, an adapted LM is built by integrating the retrieved adaptation data with background training material.

The seed text is a key aspect of this process since it is supposed to provide a good characterization of the domain in order to extract meaningful information for the adaptation. In the literature, two main approaches are commonly known: either the adaptation is supervised, i.e., the domain is known *a priori* and the considered seed text is a manually generated reliable text, typically a manual transcript [3, 6], or the adaptation is unsupervised where the seed text is obtained from ASR on spoken documents [5, 7, 8].

Obtaining large amount of seed text is desirable since large texts are assumed to more widely characterize the encountered domain. However, the feasibility of supervised adaptation depends on the size of the seed text, since the level of human effort required to produce this text manually is significant. Thus, automation of this process could provide important savings in cost and effort for the development of domain specific LMs in real-life applications.

One would naturally think that supervised approaches based on a very large seed text produce better performance than equivalent unsupervised approaches, but to the knowledge of the authors very few study has yet been conducted to verify this. Only [9] carefully examined the effect of supervision and non supervision on the performance of LM adaptation. However, the studied adaptation approach was not based on the Internet. Hence, this paper aims at comparing the Web-based domain LM adaptation process using different levels of supervision. More precisely, we seek to understand the impact of recognition errors in the seed text on speech recognition accuracy gains resulting from LM adaptation and the dependence on the size of the seed text. Since the paper focuses on LM adaptation, the problem of vocabulary adaptation is not considered here.

The paper is organized as follows: Section 2 presents the LM adaptation used in the experiments. Section 3 describes the experimental setup and introduces different adaptation scenarios for the seed text. Finally, Section 4 studies the effect of these scenarios on various aspects of our LM adaptation technique.

2. LM adaptation technique

The strategy of our LM adaptation technique is three-fold. Given a seed text which is assumed to be representative of the domain of interest, queries are first extracted. Then, Web pages are retrieved by submitting the queries to a Web search engine from which we construct an adaptation corpus. Finally, an adapted LM is trained by linearly interpolation statistics from the adaptation corpus with the set of background texts previously used to train the baseline LM. Such an adapted LM is supposed to provide higher speech recognition accuracy than the baseline LM when applied to recordings from the domain of interest. This section describes the query extraction method before explaining how Web pages are retrieved and how the adapted LM is effectively trained in our experiments.

2.1. Extracting queries from the seed text

The principle of our query extraction method, as introduced in [3], is to determine which n-grams of the baseline LM are not well enough modeled according to the given seed text and then to directly use these n-grams as queries. Given the seed text T, this principle is driven by the search for an adapted LM

This work is funded by the project CTI 12189.2 PFES-ES of the *Comité pour la Technologie et l'Innovation* (Switzerland) and was carried out in collaboration with Koemei.

whose likelihood on the seed text is greater than the one using the baseline LM, i.e.,:

$$P_A(T) > P_B(T) \; , \qquad (1)$$

where P_A and P_B respectively refer to the probability distribution of target adapted LM and of the baseline LM. This inequality can be guaranteed by decomposing the likelihood onto every n-gram (h, w) from T, where w is a word and h is a word history, leading to the following set of constraints:

$$P_A(w|h) > P_B(w|h), \qquad \forall (h, w) \in T \; . \qquad (2)$$

Then, extracting queries consists in finding out which n-grams in T are the most likely to satisfy (2). To do so, P_A can be first assumed to be a linear interpolation of P_B and probability distribution P_C trained on the corpus C of retrieved Web pages. Second, we postulate that P_C can be modeled as another linear interpolation of P_B with the probability distribution P_T trained on seed text T. Hence, (2) can be greatly simplified, as follows:

$$\lambda P_T(w|h) + (1-\lambda) P_B(w|h) > P_B(w|h), \forall (h, w) \in T \quad (3)$$
$$P_T(w|h) > P_B(w|h), \forall (h, w) \in T \; . \quad (4)$$

In practice, we approximate (4) by arbitrarily considering as queries the sole trigrams from the seed text which have not been observed during the baseline LM training, i.e., trigrams whose probability is computed by backing off. However, these n-grams may be numerous, depending on the size of the seed text T, thereby leading to a very long retrieval process and most of these n-grams are just sequences that are not specific to the domain of interest. Hence, the set of these n-grams is finally filtered by discarding any n-gram containing a stopword[1]. In our experiments, this query extraction strategy leads to a few hundred queries for a given seed text.

2.2. Web pages retrieval and adapted LM training

To retrieve domain-specific adaptation data, the queries are submitted to a Web search engine. The returned hits are downloaded following a round-robin algorithm, i.e., the i-th hits of each query are downloaded successively before downloading the $(i + 1)$-th hits, and so on. Web pages are cleaned and normalized before gathering them into an adaptation corpus. This process stops as soon as a selected number of words is reached. In our experiments, this number is set to 5 million words. On average, this threshold is reached after downloading about 20-40 pages per query.

To train the domain adapted LM, the process initially developed for the baseline LM is then re-used. More precisely, the adaptation corpus is added to the set of background corpora used to train the baseline model, and compound LMs are trained using each corpus. Then, these LMs, including the adaptation LM, are linearly interpolated such that their combination minimizes the perplexity on the seed text. Finally, the resulting LM is pruned in order to reach the same size as the baseline LM. This strategy enables to determine the relative importances of the various background corpora according to the seed. Thus, it is supposed to be better than directly linearly interpolating the baseline LM with the adaptation LM.

3. Experimental setup and adaptation scenarios

Before presenting the impact of the seed text on the adaptation process, this section presents the experimental setup, i.e., the ASR system and experimental data. Then, adaptation scenarios are introduced.

3.1. Experimental setup

The recognition system used in the experiments is a two-pass system for English. In brief, it uses individual head-mounted microphones (IHM) based acoustic models, a lexicon of $50,000$ words and a 4-gram LM trained on various corpora (AMI corpus, ICSI meeting corpus, *etc.*) for a total amount of about one billion words. The decoder is based on weighted finite state transducers. The first decoding pass relies on generic acoustic models whereas the second is performed after speaker adaptation. All details about the system architecture and the training setups can be found in [10].

The domain is represented by 57 videos produced for a university YouTube channel. While the broad domain is centered on the course content offered, these videos are of various types (faculty teaching, self-promotion, conferences, interviews, *etc.*). They have been recorded in different acoustic conditions, are of varying duration and some stakeholders are non-native English speakers. The reference transcript represents a total of $40,000$ words. The data was split into two sets: a development set of 29 videos that can be considered as the seed information source to characterize the target domain ; and a test set of 28 held-out videos. The length of the reference transcription is the same for both sets, i.e., about $20,000$ words. Out-of-vocabulary rates are $0.65\,\%$ and $0.59\,\%$ on the development set and on the test set respectively.

3.2. Adaptation scenarios

The aim of this paper is to study the importance of the seed text in achieving an effective domain LM adaptation. In fact, this adaptation may be applied within two main scenarios. Either adaptation is meant to be used in a multi-pass recognition process where spoken documents are first transcribed using the baseline LM, before adapting the LM using the first pass output as seed text with which we perform a subsequent decoding pass—we denote this as *self adaptation*. Or it is dedicated to a longer term application where the domain of documents to be transcribed in the future will remain the same—we denote this as *long term adaptation*.

Considering the development and test sets as independent, but covering the same domain, the nature of seed texts within these scenarios can vary according to two aspects: their origin and their size. Regarding the origin, the supervised case consists in considering the reference of the development set. This case is costly in terms of money and time since it requires manual transcription. Conversely, the unsupervised situation relies on the noisy transcript generated by the baseline ASR system. The word error rate (WER) of the baseline ASR is $29.6\,\%$ on the development set. Further, the levels of supervision and non supervision can be modulated by varying the seed text size. In our experiments, this is done by subsampling the seed text.

3.3. Evaluation

Effect of the domain adaptation is mainly evaluated by comparing the perplexities of the baseline LM with those of adapted LMs, on the reference transcriptions of the development set and of the test set. For most interesting settings, WERs are also reported. Results on the development set may be considered representative of a self adaptation scenario while those on the evaluation set stand for long term adaptation. Furthermore, let us notice that results for self adaptation using the reference as a seed are “cheating experiments” whose goal is to exhibit optimal (oracle) results. Finally, let us recall that no vocabulary adaptation is performed during the experiments since the paper is focusing on the sole LM adaptation task.

The next section investigates the adaptation scenarios within the two steps of the process involving the seed text.

[1] The list of stopwords is about 600 words.

4. Experiments and results

The seed text plays an important role during two steps of the domain adaptation process: it is used to extract domain-specific queries, and it helps determine the importance of the adaptation data when combining domain-specific n-gram probabilities with those obtained from the background training texts. This section thus first studies the effect of the seed text on query extraction before analyzing its role in the final linear interpolation step. Finally, the dependence on the seed text size on both steps is presented.

4.1. Effect of the seed text on query extraction

As described in Section 2, query extraction is the first step of the adaptation process. Hence, the quality of the seed text is probably crucial. To assess this hypothesis, this section compares the use of the reference and the ASR transcript of the development set ($20,000$ words each) in order to investigate the effect of recognition errors on query extraction.

Table 1 compares perplexities obtained using the baseline LM and LMs adapted from supervised and unsupervised seed texts. For every adapted LM, linear interpolation is carried out using the reference transcript in order to train optimal LMs and, thus, to highlight lower bounds of perplexity for each seed used for query extraction. It appears that, on the development set, the largest improvement is obtained when using the reference as the seed text. This is quite logical since this setting (in italic) represents an artificial case where the seed text is similar to the text modeled by the LM. It is thus common sense to observe that the improvement is less significant on the evaluation set. Interestingly, when using the ASR transcript as seed text we do not observe such differences in perplexity between the development and test data.

To better understand these first results, a second series of evaluations have been carried out whereby we isolate the correctly and incorrectly recognized parts (words) of text in the reference and in the ASR transcripts and use these sole parts as new seed texts for query extraction. Recognition errors are spotted by aligning ASR transcripts with the reference. The results of these experiments are presented in the three last rows of Table 1, where “misrecognized reference” denotes the parts of the reference which have been misrecognized using the baseline LM, “incorrect ASR” denotes what the ASR system has returned for these parts, and “correct ASR” stands for the correctly transcribed parts in the ASR. One can notice that the perplexity improvements on the development set mainly come from the misrecognized portions of the reference. This seems to be logical since it represents the word sequences which are the most inaccurately modeled by the baseline LM. However, such a conclusion is not observed on the evaluation set since the perplexity improvement obtained using “misrecognized reference” is almost the same as when only relying on the correctly recognized portions (correct ASR). Moreover, it appears that the use of “incorrect ASR” still results in perplexity improvements, though these improvements are lower. This surprising result can probably be explained by the fact that Web search engines attempt to automatically transform unlikely queries into more common word sequences while untransformed queries simply result in no hit. Further, some recognition errors may still be domain-specific words. Therefore, the use of ASR transcript is not as bad as expected since it seems that most recognition errors are harmless for query extraction, be it for long term adaptation or for self adaptation.

Table 1: *Perplexities of the development and evaluation sets using different seed texts for query extraction.*

Query extraction	Linear interp.	Dev.	Test
Baseline LM		165	170
Reference	Reference	*119*	139
ASR	Reference	133	143
Correct ASR	Reference	134	143
Incorrect ASR	Reference	142	150
Misrecognized reference	Reference	120	140

Table 2: *Perplexities on the development and evaluation sets using different texts to estimating the linear interpolation weights.*

	Query extraction	Linear interp.	Dev.	Test
	Baseline LM		165	170
(a)	Reference	Background text	159	168
	ASR	Background text	163	169
(b)	No data	Reference	154	159
	No data	ASR	155	161
(c)	Reference		119	139
	ASR		136	145
(d)	Correct ASR		135	143

4.2. Choice of the seed text for linear interpolation

The second aspect involving the seed text is the estimation of linear interpolation weights. Table 2 presents the results of experiments conducted. In addition to the seed texts previously presented, the text initially used to build the baseline LM, referred to as “background”, is introduced. As shown in rows (a), where the linear interpolation is based on the background text, it is clear that the use of adaptation data is completely inefficient if the interpolation text is disconnected from the domain. Moreover, the rows (b) show that re-interpolation of the background training texts, i.e., when no adaptation corpus is retrieved, leads to modest improvements when considering a domain-specific text to estimate the linear interpolation weights. Moreover, in this case there is nearly no difference between the use of the reference against the ASR transcript, meaning that recognition errors do not bias the interpolation weight estimation.

The set of rows (c) denotes the settings where the same text is used for both query extraction and linear interpolation, as this would probably be the case in a real application. On the whole it appears that the use of noisy seed text for interpolation as well as query generation is not significantly worse than the query generation scenario alone. Finally, the row (d) shows that by focusing on the sole correctly transcribed ASR parts linear interpolation does not perform better[2], further reinforcing previous observations. In summary it would appear that recognition errors do not bias the interpolation weight estimation (at least at the error rates that we have observed).

Achieved error rates for the settings (c) and (d) are reported in Table 3. In general, the relative trends are the same as observed for perplexity measures. More precisely, it appears that all the settings lead to significantly outperform the baseline results, even when using the ASR as a seed. Furthermore, it is clear that the recognition errors do not have any significant impact on the system performance, as was already evident from the perplexity results.

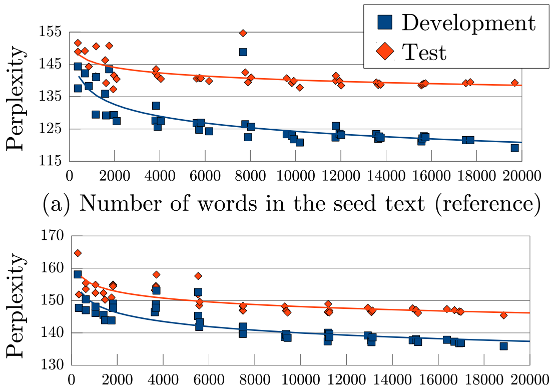
4.3. Dependence on the size of the seed text

The size of the seed text may change the conclusions drawn above concerning the low impact of recognition errors on final LM perplexities. Indeed, one would naturally assume that shorter the seed text, more variable we would expect the results of the adaptation. This is due to the fact that the domain of

[2] This is done by replacing recognition errors by out-of-vocabulary words while minimizing the perplexity of the interpolated LM.

Table 3: *WERs (%) obtained with or without domain adaptation. In brackets, relative variations w.r.t. baseline are given.*

Query extraction and linear interpolation	Development	Test
Baseline LM	29.6	25.8
Reference	26.3 (-11.1 %)	24.1 (-6.6 %)
ASR	27.3 (-7.8 %)	24.6 (-4.7 %)
Correct ASR	27.5 (-7.1 %)	24.4 (-5.4 %)



(a) Number of words in the seed text (reference)

(b) Number of words in the seed text (ASR)

Figure 1: *Perplexity of adapted LMs versus the size of the seed text by subsampling the reference (a) or the ASR transcripts (b).*

interest cannot be characterized so well. In our last series of experiments, we studied the influence of the seed text size on LM domain adaptation. Both reference and ASR transcripts from the development data were randomly subsampled on a sentence basis with different rates and these subsamples were used as new seed texts, both for query extraction and linear interpolation.

Figure 1 reports perplexities of the adapted LMs w.r.t. the size of the seed text when relying on the reference or the ASR transcripts. Firstly, it appears that the perplexity improvements decrease and their variability increases with the size of the seed text in all cases. However, this decrease is very gradual until reaching $2,000$-$4,000$ words, i.e., only 10-20 % of the original seed text size. This tends to show that the efforts spent in generating a seed text can be quite limited. Finally, it is interesting to note that the trends of the curves are the same whether the seed text is derived from the reference or from the ASR transcripts. This means that recognition errors do not appear to have strong influence on LM adaptation when reducing the seed text size.

Decoding experiments were carried out by only considering about 10-20 % of the full seed texts for LM adaptation. Resulting WERs are presented in Table 4. Regarding the reference transcriptions, WERs are quite similar to those reported in Table 3. This is very interesting from a practical point of view since it shows that in the supervised case we can annotate less data without degrading the performance. Some slight improvements even show that better adaptations can be performed with less queries, meaning that some parts of the reference are more important than others for domain adaptation. On the contrary, considering 10-20 % of the ASR transcripts leads to average increase in the WER of 0.5 % absolute compared to the use of the full development set transcript. We assume that this comes from the fact that decreasing the seed text size not only limits the ability of the text to characterize the domain but increases the impact of queries containing transcription errors. Nevertheless, WER gains w.r.t. the baseline are still significant.

Table 4: *WERs (%) obtained when reducing the size of the seed text derived from the reference or from the ASR transcripts. In brackets, relative variations w.r.t. the baseline are given.*

Query extraction and linear interpolation	Development	Test
Baseline LM	29.6	25.8
Reference (∼20 % words)	26.2 (-11.4 %)	24.4 (-5.4 %)
Reference (∼10 % words)	26.5 (-10.5 %)	24.1 (-6.6 %)
ASR (∼20 % words)	28.2 (-4.7 %)	25.0 (-3.1 %)
ASR (∼10 % words)	28.2 (-4.7 %)	24.8 (-3.9 %)

5. Conclusion

In this paper, we have conducted an investigation of supervised and unsupervised Web-based LM domain adaptation. Various scenarios have been explored to highlight the influence of the seed text used to extract queries and to perform the final linear interpolation step leading to the adapted LM. Obviously, it appears that using manual transcripts brings the greatest improvements of perplexity and ASR accuracy, but other interesting conclusions can be drawn. Firstly, the recognition errors do not bias LM adaptation, as can be seen for query extraction or for linear interpolation. This is very interesting due to the fact that error spotting in ASR outputs is a complex task. Instead, the main effect of recognition errors is a loss of information which prevents us from achieving an optimal characterization of the domain. Nevertheless, relative improvements of 7.8 % and 4.7 % over the baseline WER are achieved using the ASR transcript, depending on the adaptation scenario. Secondly, reducing the size of the seed text does not change this conclusion. Rather, the experiments have shown that decreasing the seed text size reduces both the gains in perplexity and in word error rates consistently for both supervised and unsupervised cases, though in the unsupervised case this is more pronounced.

Further aspects of supervision could be studied in the future work. For example, it would be interesting to know what is the influence of the baseline word error rate on the adaptation process. Furthermore, while having voluntarily left the problem of vocabulary adaptation aside, it would be interesting to know the influence of supervision on the recovery of domain-specific out-of-vocabulary words.

6. References

[1] X. Zhu and R. Rosenfeld, “Improving trigram language modeling with the World Wide Web,” in *Proc. of ICASSP*, 2001, pp. 533–536.

[2] A. Kilgarriff and G. Grefenstette, “Introduction to the special issue on the Web as corpus,” *Computational Linguistics*, vol. 29, no. 3, pp. 333–347, 2003.

[3] V. Wan and T. Hain, “Strategies for language model Web-data collection,” in *Proc. of ICASSP*, 2006, pp. 1520–6149.

[4] I. Bulyko, M. Ostendorf, M. Siu, T. Ng, A. Stolcke, and O. Çetin, “Web resources for language modeling in conversational speech recognition,” *ACM Trans. on Speech and Language Processing*, vol. 5, no. 1, pp. 1–25, 2007.

[5] G. Lecorvé, G. Gravier, and P. Sébillot, “An unsupervised Web-based topic language model adaptation method,” in *Proc. of ICASSP*, 2008, pp. 5081–5084.

[6] A. Sethy, P. G. Georgiou, and S. Narayanan, “Building topic specific language models from Webdata using competitive models,” in *Proc. of Eurospeech*, 2005, pp. 1293–1296.

[7] M. Suzuki, Y. Kajiura, A. Ito, and S. Makino, “Unsupervised language model adaptation based on automatic text collection from WWW,” in *Proc. of Interspeech*, 2006, pp. 2202–2205.

[8] A. Ito, Y. Kajiura, S. Makino, and M. Suzuki, “An unsupervised language model adaptation based on keyword clustering and query availability estimation,” in *Proc. of ICALIP*, 2008, pp. 1412–1418.

[9] G. Tür and A. Stolcke, “Unsupervised language model adaptation for meeting recognition,” in *Proc. of ICASSP*, 2007, pp. 173–176.

[10] T. Hain, L. Burget, J. Dines, G. Garau, M. Karafiat, D. van Leeuwen, M. Lincoln, and V. Wan, “Transcribing meetings with the AMIDA systems,” *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 20, pp. 486–498, 2012.

5.11 Paper 11: [LM12]

Conversion of Recurrent Neural Network Language Models to Weighted Finite State Transducers for Automatic Speech Recognition

Gwénolé Lecorvé, Petr Motlicek

Idiap Research Institute, Martigny, Switzerland
gwenole.lecorve@idiap.ch, petr.motlicek@idiap.ch

Abstract

Recurrent neural network language models (RNNLMs) have recently shown to outperform the venerable n-gram language models (LMs). However, in automatic speech recognition (ASR), RNNLMs were not yet used to directly decode a speech signal. Instead, RNNLMs are rather applied to rescore N-best lists generated from word lattices. To use RNNLMs in earlier stages of the speech recognition, our work proposes to transform RNNLMs into weighted finite state transducers approximating their underlying probability distribution. While the main idea consists in discretizing continuous representations of word histories, we present a first implementation of the approach using clustering techniques and entropy-based pruning. Achieved experimental results on LM perplexity and on ASR word error rates are encouraging since the performance of the discretized RNNLMs is comparable to the one of n-gram LMs.

Index Terms: Language model, recurrent neural network, weighted finite state transducer, speech decoding

1. Introduction

Recurrent neural network language models (RNNLMs) have shown to outperform the venerable n-gram language models (LMs) [1]. However, in automatic speech recognition (ASR), RNNLMs cannot be used to directly decode a speech signal since they rely on continuous representations of word histories while decoding algorithms require to handle discrete representations to remain tractable [2, 3]. Instead, RNNLMs are currently used to rescore N-best lists generated using n-gram LMs. Hence, the prediction power of RNNLMs is used only on subsets of all transcription hypotheses. Such an approach does not offer the optimal solution since the n-gram LM used for the decoding may have discarded hypotheses which the RNNLM would have judged very likely. Furthermore, the distributions of these two kinds of LMs have been shown to be complementary [4, 5]. The use of RNNLMs in early stages of speech decoding is thus a challenging objective.

Recently, few studies were devoted to this problem. In [6], the authors propose to sample word sequences by using RNNLM as a generative model before training an n-gram LM based on the generated text. By exploiting this LM to perform first pass decoding, achieved results outperformed the use of n-gram LMs trained on standard texts. However, we assume that this approach is still not optimal since it still prevents from relying on long span information during the decoding. In [7], the author has proposed an *iterative decoding* algorithm which enables to efficiently rescore word lattices using RNNLMs. The main idea is to partition word lattices to reduce the computational complexity of browsing all possible hypotheses. Though leading to good results, this technique cannot be directly applied in the first pass of the decoding since no explicit search graph is available at this moment of the recognition process.

In this paper, we define a new generic strategy to transform RNNLMs into a Weighted Finite State Transducer (WFST) which can directly be used within the decoding process of an ASR system [3]. We believe that this approach has a potential to outperform the conventional approach where RNNLMs are employed to rescore N-best hypotheses as a final step of ASR. The principle of the conversion consists in discretizing continuous RNNLM representations of word histories in order to build WFST states, and then to link these states with probabilities derived from the RNNLM. In practice, this approach also raises some needs for pruning the generated WFST since the theoretical number of states may be large according to the chosen discretization strategy. We present a preliminary implementation of the RNNLM conversion algorithm based on K-means clustering and entropy pruning.

This paper is organized as follows: after recalling the principles of RNNLMs provided in Section 2, the generic conversion strategy is introduced in Section 3. Section 4 presents how K-means clustering and entropy pruning can be used to implement a first version of the generic strategy. Finally, Section 5 describes experiments on the Penn Treebank corpus and using LVCSR meeting system.

2. Overview of RNNLMs

The goal of a language model employed in ASR system is to provide the conditional probability of a word w_i given an history h of preceding words. As detailed in [1], this history is represented in RNNLMs by the most recent preceding word w_{i-1} and a multi-dimensional continuous representation $\mathbf{c}_i$ of the remaining context. The topology of the neural network used to compute conditional probabilities $P[w_i|w_{i-1}, \mathbf{c}_{i-1}]$ is organized in 3 layers using a bottleneck scheme. The input layer reads a word w_{i-1} and a continuous history $\mathbf{c}_{i-1}$. The hidden layer compresses the information of these two inputs and computes a new representation $\mathbf{c}_i$. The value $\mathbf{c}_i$ is then passed to the output layer which, after normalization, provides the conditional probabilities.

3. Generic RNNLM conversion

The goal of this work is to convert RNNLMs into an approximate WFST representation. As illustrated in Figure 1, this task mainly consists in binding discrete states with the continuous input states of the RNNLM, and in using these discrete states as the nodes of a WFST. The edges among nodes are then labeled with word transitions and their probabilities estimated by the RNNLM. There are two key aspects to achieve this task. First, a *discretization function* needs to be defined to transform the continuous representations. Second, we have to take into account the size of the output WFST since enumerating all possible discrete states might quickly be untractable as soon as the vocabulary becomes large and the discretization becomes precise. Thus, a *pruning criterion* needs to be defined in order to discard uninteresting discrete states, and a *back-off strategy* in order to model the pruned states in a simpler way. This sec-

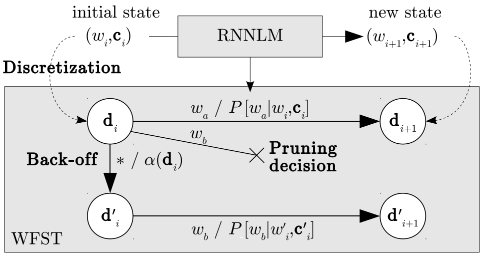


Figure 1: *Overview of the RNNLM discretization scheme.*

tion formally introduces these parameters before providing the generic algorithm for the conversion.

3.1. Discretization function

The main function to be defined is a discretization function which returns a discrete state for every possible input continuous state $(w_i, \mathbf{c}_i)$. The generic form of this discretization function f is as follows:

$$f: \begin{array}{ccccc} V & \times & \mathbb{R}^h & \longrightarrow & \mathbb{N}^k \\ w_i & , & \mathbf{c}_i & \longmapsto & \mathbf{d}_i \ , \end{array} \qquad (1)$$

where V is the LM vocabulary, h is the size of the hidden layer of the RNNLM, and k is the dimension of the discrete representation $\mathbf{d}_i$. As it will be shown in the algorithm, we also need to be able to go back from discrete states to continuous representations. Thus, the partial inverse function f^{-1} must also be defined such that:

$$f^{-1}: \begin{array}{ccc} \mathbb{N}^k & \longrightarrow & Y \subset V \times \mathbb{R}^h \\ \mathbf{d}_i & \longmapsto & (w_i, \mathbf{c}_i) \ . \end{array} \qquad (2)$$

This function is only a partial inverse of function f since f is, by definition, a surjection. Thus, the codomain of f^{-1} is limited to a subset of $V \times \mathbb{R}^h$.

3.2. Pruning criterion and back-off

Since the WFST is intended to be used in ASR during the decoding, it should not be too large. Thus, it is important to be able to control the size of the WFST by pruning and by introducing a back-off scheme. Furthermore, since the cardinality of the discrete domain $\mathbb{N}^k$ can be huge, the pruning should be done on the fly, i.e., while building the WFST. This requires to define two parameters.

First, a pruning criterion π must be defined to decide whether a corresponding edge should be discarded or added when building the WFST. A "good" pruning criterion should be such that the pruning of an edge does not lead to large information loss. Given a node $\mathbf{d}_i$, the criterion should thus be derived from the quantity of information carried by a new word transition v. The generic form of this criterion is

$$\pi(v, \mathbf{d}_i) = \begin{cases} \textit{false} & \text{if } P(v|f^{-1}(\mathbf{d}_i)) \text{ is informative enough,} \\ \textit{true} & \text{otherwise .} \end{cases} \qquad (3)$$

Second, a back-off mechanism must be introduced in order to approximate the probability of pruned events[1]. Basically, this strategy requires to define a back-off function β which maps a given discrete representation to a simpler representation:

$$\beta: \begin{array}{ccc} \mathbb{N}^k & \longrightarrow & \mathbb{N}^k \\ \mathbf{d}_i & \longmapsto & \mathbf{d}'_i \ . \end{array} \qquad (4)$$

```
Data: $L$, a list of discrete states, i.e., of WFST nodes
$L \leftarrow f$(beginning of sentence);
while $L \neq \emptyset$ do
    $\mathbf{d}_{\text{src}} \leftarrow$ pop($L$);
    $(w_{\text{src}}, \mathbf{c}_{\text{src}}) \leftarrow f^{-1}(\mathbf{d}_{\text{src}})$;
    $\mathbf{c}_{\text{dst}} \leftarrow$ hidden_layer($w_{\text{src}}, \mathbf{c}_{\text{src}}$);
    foreach $v \in V$ do
        if $\mathbf{d}_{src} = \mathbf{d}_{min}$
        or not $\pi(v, \mathbf{d}_{\text{src}})$ then
            $p \leftarrow P(v|w_{\text{src}}, \mathbf{c}_{\text{src}})$;
            $\mathbf{d}_{\text{dst}} \leftarrow f(v, \mathbf{c}_{\text{dst}})$;
        else
            $p \leftarrow 0$ ;
            $v \leftarrow \epsilon$;
            $\mathbf{d}_{\text{dst}} \leftarrow \beta(\mathbf{d}_{\text{src}})$;
        if node $\mathbf{d}_{\text{dst}}$ does not exist then
            add_node_to_wfst($\mathbf{d}_{\text{dst}}$);
            push($L, \mathbf{d}_{\text{dst}}$);
        add_edge_to_wfst($\mathbf{d}_{\text{src}} \xrightarrow{v:v,p} \mathbf{d}_{\text{dst}}$);
compute_backoff_weights();
```

Algorithm 1: *Pseudo-code of the RNNLM conversion.*

The destination state $\mathbf{d}'_i$ is referred to as the *back-off state* or *node* of the state $\mathbf{d}_i$. To avoid cycles in the WFST, the function β must define a partial order over all discrete states, i.e., it must guarantee that $\mathbf{d}'_i$ is "simpler" than $\mathbf{d}_i$. This naturally introduces the notion of minimal state $\mathbf{d}_{\text{min}}$, i.e., the state for which no pruning and back-off can be done.

3.3. Conversion algorithm

Assuming that the discretization function f, the pruning criterion π, and the back-off function β are defined, the conversion algorithm seeks to discretize each given RNNLM state and to build outgoing edges reaching new states. This process can be written in an iterative way whose pseudo-code is given by the Algorithm 1. Given the list of states which have already been added to the WFST but for which no outgoing edge has been created yet, the algorithm pops a state, computes probabilities using the RNNLM, and then builds edges. As soon as an edge reaches a new destination node in the WFST, this next node is built and pushed into the list on remaining states to be explored. In practice, the conversion process starts with the RNNLM state corresponding to a beginning of sentence. When considering pruning, a decision must be made before adding a new edge. If the edge starts from the minimal state or carries enough information, then it is built. Otherwise, it is discarded and redirected to a back-off node. The weights of back-off transitions are computed after building the WFST by following Katz's strategy [8].

4. Implementation

We propose to implement the generic process described above by using K-means clustering for the discretization of RNNLM states and entropy-based criteria for the pruning strategy. This implementation is described in this section.

4.1. Discretization using K-means

We propose to discretize RNNLM states by first partitioning their continuous domain into clusters computed using the K-means algorithm, and then by associating each state to a corresponding cluster. Each cluster is denoted by an identifier and is associated with its centroid. Given a set of K centroids, the

[1] Even if an event is judged as unlikely, it does not mean that it cannot occur. Hence, the model must be able to provide a probability for any event.

discretization and "undiscretization" functions are defined as:

$$f_K(w\ ,\ \mathbf{c}) = (\ w\ ,\ k\ )\ , \tag{5}$$

where $k \in [\![1, K]\!]$ is the ID of the centroid associated with $\mathbf{c}$, and

$$f_K^{-1}(w\ ,\ k) = (\ w\ ,\ \mathbf{c}_k\ )\ , \tag{6}$$

where $\mathbf{c}_k$ is the k-th centroid. As mentioned in Section 3.1, we can clearly see that, in most cases, $f_K^{-1}\big(f_K(x)\big)$ does not equal to x, which means that some information is lost.

An advantage of K-means is that the size of the discrete space of the WFST nodes can be explicitly set through K. Nonetheless, for a large vocabulary, the size of the final WFST can be huge if no pruning is applied[2]. To train the centroids, the RNNLM is first applied on a text data, e.g., the training text. Then, each continuous state $\mathbf{c}_i$ encountered is stored and the K-means clustering is performed on these logs.

4.2. Back-off

We define a two-fold back-off scheme such that the information about the long-span context is dropped as first and the information about the last word is dropped as second. Formally, given a discrete state (w, k), this consists in defining $\beta(w, k) = (w, \varnothing)$ and $\beta(w, \varnothing) = (\varnothing, \varnothing)$ where $\varnothing$ means that no information is provided. To remain compliant with the method, values are defined according to f^{-1} for these two special discrete states: $f^{-1}(w, \varnothing) = (w, \mathbf{c}_0)$ where $\mathbf{c}_0$ is the global mean of all the continuous states observed during the K-means clustering, and $f^{-1}(\varnothing, \varnothing) = (\varnothing, \mathbf{c}_0)$.

4.3. Pruning

Within the conversion process, the principle of pruning is to reduce the final WFST size by not building edges whose absence does not result in significant information loss. A well known strategy for this problem consists in identifying edges which have a minimal effect on the entropy of the probability distribution [9]. Following this principle, we define our pruning criterion based on two values.

First, the piece of entropy carried by a transition from the state $\mathbf{d}$ with the word v is considered. It is expressed as:

$$H(v, \mathbf{d}) = -P(v, f_K^{-1}(\mathbf{d})) \cdot \log P(v, f_K^{-1}(\mathbf{d}))\ . \tag{7}$$

By denoting $P(v|f_K^{-1}(\mathbf{d}))$ to $P(v|\mathbf{d})$, the joint probability $P(v, f_K^{-1}(\mathbf{d}))$ of a word v and its history $\mathbf{d}$ can be approximated as:

$$\begin{aligned} P(v, f_K^{-1}(\mathbf{d})) &\approx P(v|\mathbf{d}) \cdot P(\mathbf{d}) & (8)\\ &= P(v|w, \mathbf{c}_k) \cdot P(w, k)\ . & (9) \end{aligned}$$

The probability $P(v|w, \mathbf{c}_k)$ is directly given by the RNNLM while the probability $P(w, k)$ is considered as a prior estimated from the logs used to train the centroids. Since the estimation of this joint probability may be unreliable because of data sparsity, an independence assumption between w and k is made. In practice, $P(w, k)$ is thus simplified to $P(w) \cdot P(k)$.

Second, an important aspect is to know if the probabilities of an event remain close before and after pruning. For a transition $(v, \mathbf{d})$, the relative difference between these two probabilities is defined as:

$$D(v, \mathbf{d}) = \frac{|P(v|\mathbf{d}) - \alpha(\mathbf{d}) \cdot P(v|\beta(\mathbf{d}))|}{P(v|\mathbf{d})}\ , \tag{10}$$

Table 1: *Perplexities of n-gram LMs and of the RNNLM.*

2-grams	3-grams	4-grams	5+-grams	RNNLM
186	148	142	141	124

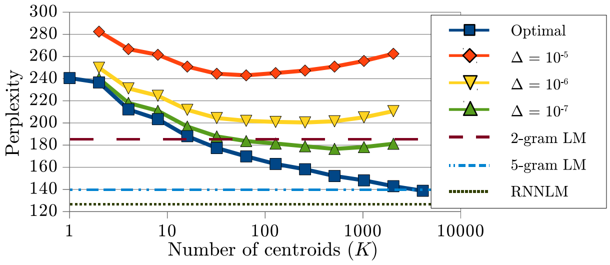


Figure 2: *Perplexities on test set of the Penn Treebank corpus for WFSTs generated using various numbers of centroids and various pruning thresholds Δ.*

where $\beta(\mathbf{d})$ is the back-off state for $\mathbf{d}$, and the back-off weight $\alpha(\mathbf{d})$ is approximated by iteratively estimating the probability mass of events which will be pruned for the state $\mathbf{d}$. The higher $D(v, \mathbf{d})$, the lesser backing off preserves the original probability.

Finally, for a node $\mathbf{d}$ and a transition word v taken under examination, we define the pruning criterion as follows:

$$\pi(w, \mathbf{d}) = \begin{cases} \textit{false} & \text{if } H(v, \mathbf{d}) \cdot D(v, \mathbf{d}) < \Delta \\ \textit{true} & \text{otherwise}\ , \end{cases} \tag{11}$$

where Δ is a user-determined pruning threshold (the lower the value, the larger the size of the WFST).

The whole process has been implemented using the RNNLM toolkit[3] and the OpenFst library[4]. Conversions last between a few minutes and a few hours according $|V|$, K, and Δ.

5. Experiments

Two series of experiments have been carried out to evaluate the proposed approach: (a) experiments on the Penn Treebank corpus to study the behavior of the conversion process, and (b) the decoding experiments on meeting speech recordings using a large vocabulary continuous speech recognition system.

5.1. Perplexities on the Penn Treebank corpus

The goal of the first part of experiments is to study an impact of the K-means algorithm as well as the pruning threshold on the RNNLM conversion. To do so, we use the same LMs as those used in [1] on the Penn Treebank corpus. Two types of LMs are considered: n-gram LMs trained with various orders using maximum likelihood estimation and Kneser-Ney smoothing, and a RNNLM based on a hidden layer of 300 neurons. The Penn Treebank corpus is a portion of the Wall Street Journal which is widely used for evaluating performance of statistical LMs [10]. This corpus is split into 3 parts: a training set of 900K words, a development set of 70K words (which is only used for RNNLM training), and a test set of 80K words. The vocabulary is made of 10K words. Perplexities of these models on the test set are reported in Table 1.

Various values of K have been used to extract centroids from the training set, as described in Section 4.1. Furthermore, 3 different values have been set for the pruning threshold. WFSTs are generated using these settings and the final perplexities are measured on the test set. These perplexities are reported in Figure 2 and are compared with those of the other models.

[2] Precisely, the theoretical maximum numbers are $|V| \times K$ nodes and $(|V| \times K)^2$ edges.

[3] http://www.fit.vutbr.cz/~imikolov/rnnlm/

[4] http://www.openfst.org

Table 2: *Perplexities of LMs on the evaluation set of RT 2007.*

2-gram LM	4-gram LM	WFST	RNNLM
162	93	127	93

Additionally, the optimal perplexity, which can be obtained if no pruning was applied, is given in Figure 2. First, it appears that this optimal value decreases when the number K of centroids increases (as increasing K means that richer history can be considered). Although the optimal perplexity does not reach the perplexity of the original RNNLM, these preliminary results interestingly show that the discretization does not lead to large information loss as long as K is large enough. Then, a degradation can clearly be observed when introducing pruning (i.e., $\Delta > 0$), which is obvious since most of the possible transitions are pruned[5]. These degradations increase as K becomes too large, which probably highlights some weaknesses of our current implementation. This can mainly be explained by the fact that the average prior probability of any centroid decreases as K increases. This leads to reduce the number of transitions which are informative enough according to the pruning threshold. Moreover, this phenomenon can become worse by taking into account more unreliable prior probabilities of centroids for high values of K because of the limited size of the training set.

5.2. Decoding of meeting data

Second, preliminary decoding experiments have been carried out on the evaluation set of NIST RT 2007[6] dataset (35K words). We use a two-pass recognition process where word lattices are first generated using “simple” models, leading to N-best lists, with N set to $1,000$. Then, more complex LMs are used to rescore these lists. For the rescoring, we use the RNNLM described in [5] and a 4-gram developed for the AMI system [11]. The RNNLM has been trained on 26.5M words with a $65K$ words vocabulary while the n-gram LM is trained on about a billion words with the same vocabulary. Both models reach the same perplexity on the evaluation set of RT 2007. For the decoding, a WFST is built from the RNNLM based with $K = 512$ and $\Delta = 10^{-7}$ and a bigram LM is derived from the 4-gram LM. The WFST and the bigram LM are about the same size. Perplexities of all LMs on RT 2007 are given in Table 2. Acoustic model is represented by relatively simple HMM/GMM trained using maximum likelihood over PLP features (39 dimensions). The model contains 4.5K tied states with 18 Gaussian mixture components per state. No speaker adaptation is performed in order to keep reasonable run times.

Table 3 reports the word error rates (WER) of the best hypothesis directly after the decoding pass using the bigram LM or the WFST, and after rescoring with the 4-gram LM or with the RNNLM. Additionally, the WERs of the best hypothesis returned without any rescoring, i.e., by using only the WFST and the bigram LM, are given. First, we can notice that WERs are a bit high. This is due to the absence of speaker adaptation. Then, it appears that the WER obtained using the WFST is better than when using the bigram LM since an absolute difference of $0.5\,\%$ is reported, as this was suggested by the perplexities. This is consistent with observed perplexities. Finally, after rescoring, the difference is lesser when using the 4-gram LM and it is even reversed when using the RNNLM. Nonetheless, these results are encouraging since our preliminary implementation of the RNNLM conversion scheme performs already as well as n-gram LMs. We will thus continue experiments.

[5] For instance, for $K = 2$ and $\Delta = 10^{-5}$, $99.946\,\%$ of the transitions are pruned, and, for $K = 1024$ and $\Delta = 10^{-7}$, this number becomes $99.986\,\%$.

[6] http://www.itl.nist.gov/iad/mig/tests/rt/2007/

Table 3: *WERs on the evaluation set of RT 2007 using n-gram LM or RNNLM-derived WFST to generate N-best lists and using n-gram LM or RNNLM to rescore them.*

Decoding Rescoring	2-gram LM	WFST derived from RNNLM
No rescoring	47.8 %	47.3 %
4-gram LM	45.2 %	45.0 %
RNNLM	42.9 %	43.2 %

6. Conclusion

In this paper, we have proposed a new strategy to directly exploit probabilities estimated by RNNLMs in the ASR decoder. This strategy consists in converting a RNNLM into a WFST by means of discretization and pruning. We have proposed an original implementation of this generic strategy by using K-means clustering and entropy-based pruning. Achieved results on the Penn Treebank and RT 2007 corpora show that this strategy is promising since the generated WFSTs lead to similar performance to the one of n-gram LMs. Nevertheless, some improvements are still necessary. Especially, a more elaborate pruning criteria could be defined to examine the importance of a transition. However, this task is difficult since estimating the entropy of a RNNLM is complex. Finally, the discretization step could probably also be improved. For instance, it could be interesting to use other possible distances than the default L2 distance to compute the centroids. Measures based on the Kullback-Leibler divergence appear as a natural option towards this objective. Eventually, the employment of hierarchical clustering may also reduce the loss of information due to back-off.

7. Acknowledgements

The authors would like to thank Tomáš Mikolov for providing the RNNLMs used in this paper. This work is funded by the project CTI 12189.2 PFES-ES of the *Comité pour la Technologie et l’Innovation* (Switzerland).

8. References

[1] T. Mikolov, M. Karafiat, L. Burget, J. Černockỳ, and S. Khudanpur, “Recurrent neural network based language model,” in *Proc. of Interspeech*, 2010, pp. 1045–1048.

[2] H. Ney and S. Ortmanns, “Dynamic programming search for continuous speech recognition,” *IEEE Signal Processing Magazine*, pp. 64–83, 1999.

[3] M. Mohri, F. C. Pereira, and M. Riley, “Speech recognition with weighted finite-state transducers,” *Springer Handbook of Speech Processing*, pp. 559–584, 2008.

[4] T. Mikolov, A. Deoras, S. Kombrink, L. Burget, and J. Černocký, “Empirical evaluation and combination of advanced language modeling techniques,” in *Proc. of Interspeech*, 2011, pp. 605–608.

[5] S. Kombrink, T. Mikolov, M. Karafit, and L. Burget, “Recurrent neural network based language modeling in meeting recognition,” in *Proc. of Interspeech*, 2011, pp. 2877–2880.

[6] A. Deoras, T. M. Mikolov, S. Kombrink, M. Karafiát, and S. Khudanpur, “Variational approximation of long-span language models for LVCSR,” in *Proc. of ICASSP*, 2011, pp. 5532–5535.

[7] A. Deoras, “Search and decoding strategies for complex lexical modeling in LVCSR,” Ph.D. dissertation, Johns Hopkins University, 2011.

[8] S. M. Katz, “Estimation of probabilities from sparse data for the language model component of a speech recognizer,” *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 35, no. 3, pp. 400–401, 1987.

[9] A. Stolcke, “Entropy-based pruning of backoff language models,” in *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, 1998, pp. 270–274.

[10] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, “Building a large annotated corpus of English: the Penn Treebank,” *Computational Linguistics*, vol. 19, no. 2, pp. 313–330, 1993.

[11] T. Hain, L. Burget, J. Dines, P. N. Garner, A. E. Hannani, M. Huijbregts, M. Karafit, M. Lincoln, and V. Wan, “The AMIDA 2009 meeting transcription system,” in *Proc. of Interspeech*, 2010, pp. 358–361.

5.12 Paper 12: [Nig+21]

A TWO-STEP APPROACH TO LEVERAGE CONTEXTUAL DATA: SPEECH RECOGNITION IN AIR-TRAFFIC COMMUNICATIONS

Iuliia Nigmatulina [†,‡], *Juan Zuluaga-Gomez* [†,§], *Amrutha Prasad* [†,¶], *Seyyed Saeed Sarfjoo* [†], *Petr Motlicek* [†]

[†] Idiap Research Institute, Martigny, Switzerland
[‡] Institute of Computational Linguistics, University of Zürich, Switzerland
[§] Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland
[¶] Brno University of Technology, Brno, Czech Republic

ABSTRACT

Automatic Speech Recognition (ASR), as the assistance of speech communication between pilots and air-traffic controllers, can significantly reduce the complexity of the task and increase the reliability of transmitted information. ASR application can lead to a lower number of incidents caused by misunderstanding and improve air traffic management (ATM) efficiency. Evidently, high accuracy predictions, especially, of key information, i.e., callsigns and commands, are required to minimize the risk of errors. We prove that combining the benefits of ASR and Natural Language Processing (NLP) methods to make use of surveillance data (i.e. additional modality) helps to considerably improve the recognition of callsigns (named entity). In this paper, we investigate a two-step callsign boosting approach: (1) at the 1st step (ASR), weights of probable callsign n-grams are reduced in G.fst and/or in the decoding FST (lattices), (2) at the 2nd step (NLP), callsigns extracted from the improved recognition outputs with Named Entity Recognition (NER) are correlated with the surveillance data to select the most suitable one. Boosting callsign n-grams with the combination of ASR and NLP methods eventually leads up to 53.7% of an absolute, or 60.4% of a relative, improvement in callsign recognition.

Index Terms— automatic speech recognition, human-computer interaction, Air-Traffic Control, Air-Surveillance Data, Callsign Detection, finite-state transducers

1. INTRODUCTION

Key components of speech communication between pilots and Air-Traffic Controllers (ATCo), i.e., callsigns, which are used for identification of aircrafts, and providing commands, demand high recognition accuracies. Callsigns are unique identifiers for aircrafts, of which the first part is an abbreviation of airline name and the last part is a flight number that contains a digit combination and may also incorporate an additional character combination, e.g., *TVS84J* (see Table 1). At a certain time point, only few aircrafts are usually in the radar zone which means only a limited number of callsigns can be referred to in the ATCo communications. If a recognized callsign does not match any 'active' callsign registered by radar at the given time point, it means that there is no corresponding aircraft in the air space and the automatically recognized command (from voice communication) is invalid. Therefore, contextual information coming from the surveillance (radar) data allows adjusting system predictions that can significantly increase its accuracy.

Table 1. Callsigns: compressed and extended (airlines designators are in bold)

Callsign	Extended callsign
SWR2689	**swiss** two six eight nine
RYR1RK	**ryanair** one romeo kilo
RYR1SG	**ryanair** one sierra golf

Although contextual information has been already used in previous ATC studies [1–4], or more recently in [5–7]; it has been never adapted for both ASR and concept extraction outputs simultaneously and without a need of any additional knowledge (e.g., manual annotation, classes, etc.). This research aims to leverage the available contextual information by combining ASR and NLP methods. We believe that ASR and NLP are complementary tasks rather than separated ones. Whereas ASR exploits speech to produce a sequence of words, NLP exploits the intrinsic characteristics in a given snippet of text. ASR normally struggles to model long sequences, while state-of-the-art NLP systems allow extracting key information in the whole chunks of text; for instance an entire ATC utterance. In the proposed approach, we focus on an iterative use of contextual data, to take advantage of a combination of ASR and NLP modules. (1) First, boosting the probability of active callsigns in ASR system (*FST-boosting*), (2) second, boosting ASR outputs (*NLP-boosting*) in order to correct those predicted callsigns, which are not present in the surveillance data.

The rest of the paper is organised as follows: Section 2 reviews current approaches on integrating contextual knowledge in ASR for ATC communications. Section 3 gives a theoretical background of the proposed ASR-NLP approach to leverage surveillance data. Then, we present the data and the experiment set up in Section 4. Finally, we report the results and summarise our observations and ideas in Section 5 and 6, respectively.

2. CONTEXTUAL INFORMATION FOR CALLSIGN DETECTION

Contextual data on the ASR level can be integrated by modifying weights of target n-grams in the grammar or/and in the ASR decoding lattices, e.g. by mean of generalised composition of baseline

The work was supported by the European Union's Horizon 2020 projects. No. 864702 - ATCO2 (Automatic collection and processing of voice data from air-traffic communications), which is a part of Clean Sky Joint Undertaking. The work was partially supported by SESAR Joint Undertaking under Grant Agreement No. 884287 - HAAWAII (Highly automated air traffic controller workstations with artificial intelligence integration).

LM and Weighted Finite State Transducers (WFSTs) with the target contextual n-grams [8–10]. A similar approach has been recently adopted in the ATC domain [5, 6] and proved to offer a significant gain in callsign recognition. A list of callsigns to be boosted is regularly changing and needs to be updated dynamically per each utterance. Thus, weights of callsign n-grams are dynamically modified in the WFST. The first of the methods is lattice rescoring, where the weights are adjusted on the word recognition lattices from the first pass decoding. In the other method, weights are dynamically modified directly in the grammar (G.fst), which allows having target n-grams boosted before the decoding is performed [6]. For our experiments, we will adopt the lattice rescoring approach to leverage the performance on the ASR side.

Besides the ASR performance, contextual information for ATC has been also used to improve concept extraction [1–4]. Schmidt et al. [1] applied a Context-Free Grammar (CFG)-based LM limiting the search space according to the contextual data. Shore et al. [2] and Oualil et al. [3, 4] build a CFG-based concept extractor with all semantic concepts of ATC embedded in XML annotation tags. In [2], after decoding, the lattice hypotheses are rescored by incorporating an additional knowledge source component to the cost function. The knowledge-based rescoring penalises hypotheses that are invalid in the context, e.g., callsigns not registered in the air space. In [3], to overcome the problem of variability of ATCO commands, the weighted Levenshtein distance is applied to find the closest match between an ASR hypothesis and generated context word sequences. [4] combines methods from [2, 3] adding more contextual constraints from data with temporal information. Although these methods help to considerably increase the recognition accuracy, their limitation is that it deals only with concepts and callsigns which are annotated and included into the grammar. Those n-grams that do not appear in the grammar can not be extracted and evaluated. Finally, Helmke et al. [11] recently proposed a machine learning algorithm for command extraction from the ASR hypothesized outputs with the use of keywords. This model achieves good results and it is the second alternative approach to our methods.

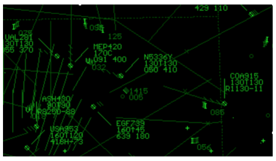
3. METHODS

We focus on the combination of ASR and NLP methods and investigate two-steps approach for callsigns extraction. As a callsign is a sequence of words, using contextual information to improve recognition of callsigns is a task of boosting n-grams. The contextual data comes from radar in a compressed form, i.e., standardized phraseology format of International Civil Aviation Organization (ICAO) [12] (see Fig. 1). To introduce the contextual knowledge into the ASR system, all callsigns need to be expanded to word sequences (Table 1). The compressed form often allows more than one possible realisation in the ATCos' speech: For example, **DLH5KX** can be expanded as *'hansa five kilo x-ray'* or *'lufthansa five kilo x-ray'*, etc. As we can not say which particular expansion is true for an uttered callsign, it is important to take all expansion variants into account.

3.1. Integration of contextual knowledge into ASR system

In a standard hybrid-based ASR system, the different knowledge sources are represented as WFSTs, which are combined by the 'composition' operator together in the final decoding graph [13]. Information from additional knowledge sources can be also integrated into a system by means of composition.

Our first integration of contextual knowledge into ASR is done on the LM level (*G-extension*). The idea is to boost callsign n-grams



AL78 ACA854 AFR6735 BEE6BU CLX79W CLX98G DAL32 DLH6LY EIN23W EXS155 EXS95LP EXS9DL EZY61MT EZY82NY HMD61 ICE416 RYR1RK RYR3TA RYR4DD RYR4TM RYR50HD RYR7245 SAS99K STK19L STK21GL STK21M STK31S STK3212

Fig. 1. Callsigns in ICAO format received from radar.

already available in LM, and even more important to add those callsign n-grams, which are absent (e.g., >3 words sequences in 3-gram LM). We build a contextual *FST* that includes all possible callsigns from the tower: all callsigns registered by the radar at different time stamps (from 17K to 280K callsigns to boost in different test sets; see last column in Table 2). Then, the main $G.fst$ is composed with the contextual $G\_biased.fst$ and the result of composition is used in the final decoding $HCLG$ graph.

The second integration of contextual information (*lattice rescoring*) is done per utterance on top of the decoding lattices which allows flexible adaptation to new-coming contextual information avoiding changing the main decoding graph ($HCLG$) (for more details check [6]). Weights in lattices are rescored according to the surveillance data: for each test utterance, an FST biased to callsigns n-grams registered at the time stamp when an utterance is created and composed with lattices created in the first pass:

$$Lattices' = Lattices \circ biasing\_FST \qquad (1)$$

Weights updated in the composition are used for final predictions.

3.2. Integration of contextual knowledge on ASR transcripts

Our approach for integrating contextual knowledge on ASR transcripts (e.g., 1-best hypothesis) is based on a two-step pipeline. Each step conveys an independent module.

3.2.1. Named Entity Recognition (NER) module

ATC communications carry rich information such as callsigns, commands, values and units; they can be seen as 'named entities'. We propose a NLP-based system to extract such information from ASR transcripts. We defined callsigns, commands, units, values, greetings OR the rest (e.g., 'None' class) as tags for the NER task, as depicted in Figure 2. First, we downloaded a BERT [14] model pre-trained as masked language model from Huggingface [15] and fine-tuned it on NER task with 12k sentences (∼12 hours of speech), where each word has a tag. Then, we developed a data augmentation pipeline in order to increase the amount of training data: 1M samples from 12k sentences. The pipeline has four actions that modifies the training sample: *add*, *delete*, *swap*, or *move* the **callsign** across the utterance -sentence-. *Delete* and *move* actions, remove and keep the same callsigns, respectively; *add* and *swap* generate a sentence with a new callsign picked randomly from a callsign list. The callsign list is pre-defined by a user, which makes the approach easy to deploy in out-of-domain data (i.e., callsigns from different airports/countries).

3.2.2. Re-ranking module based on Levenshtein distance

The BERT-based system for NER allows us to extract the callsign from a given transcript or ASR 1-best hypotheses. Recognition of this entity is crucial where a single error produced by the ASR system affects the whole entity (normally composed of three to eight

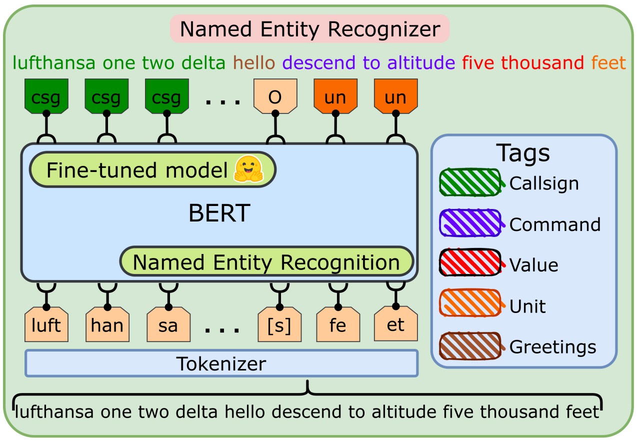


Fig. 2. BERT-based model (Huggingface) fine-tuned on NER task.

words). Additionally, speakers regularly shorten callsigns in the conversation making it impossible for an ASR system to generate the full entity (e.g., *'three nine two papa'* instead of *'austrian three nine two papa'*, *'six lima yankee'* instead of *'hansa six lima yankee'*). One way to overcome this issue is to re-rank entities extracted by the BERT-based NER system with the surveillance data. The output of an NER system is a list of tags that match words or sequences of words in an input utterance. As our only available source of contextual knowledge are callsigns registered at a certain time and location, we extract callsigns with the NER system and discard other entities. Correspondingly, each utterance has a list of callsigns expanded into word sequences (shown in Table 1). As input, the re-ranking module takes (i) a callsign extracted by the NER system and (ii) an expanded list of callsigns. The re-ranking module compares a given n-gram sequence against a list of possible n-grams, and finds the closest match from the list of surveillance data based on the weighted Levenshtein distance. We skip the re-ranking in case the NER system outputs a 'NO_CALLSIGN' flag (no callsign recognized).

4. DATA AND EXPERIMENTAL SETUP

4.1. Data

For the callsign boosting experiments, we use four test sets; all of them have utterances both with and without callsigns (see Table 2).

LiveATC: the first test set is from the LiveATC[1] data recorded from publicly accessible VHF radio channels, which includes both pilots and ATCo speech and, therefore, is of rather low quality (i.e., low SNR often below 10dB) [16].

MALORCA: Prague and Vienna test sets are mainly of good quality (i.e., telephone quality speech with SNR usually above 20dB) data from the MALORCA project [17, 18][2] which includes only ATCo speech. The recognition accuracy of the baseline model are already high above the one reached on VHF LiveATC data (see Table 3). The data was collected from the Prague and Vienna airports and, thus, forms two separate sets correspondingly.

NATS: a data set collected under HAAWAII project[3] with the data coming from London approach (airport). This data is relatively high-quality, similar to MALORCA.

Table 2. Test sets (callsigns (csgn) per utterance (utt) — median of callsigns per utterance in the surveillance data)

Test set	N of utt with	 w/o	Csgn per utt	Min	All csgns
	a csgn				
LiveATC	581	29	28	40	280K
Malorca Prague	784	88	5	82	17K
Malorca Vienna	877	38	19	65	59K
NATS	794	73	50	50	168K

The data sets are used differently in training ASR and NER models. The ASR train data includes Malorca sets but not LiveATC and NATS. The data for fine-tuning the NER system contains LiveATC data but neither Malorca, nor NATS sets.

4.2. ASR model

For training the baseline acoustic model, as well as for the decoding and rescoring experiments, we used the Kaldi framework [19]. The system follows the standard Kaldi recipe, which uses MFCC and i-vectors features. The standard chain training is based on Lattice-free MMI (LF-MMI) [20], which includes 3-fold speed perturbation and one third frame sub-sampling.

The acoustic model is a CNN-TDNNF trained on approximately 1200 hours of ATC labeled augmented data [16, 21]. First, the training databases (195 hours[4]) were augmented by adding noises that match LiveATC audio channel (one batch between 5-10 dB and other 10-20dB SNR). Afterwards, we applied speed perturbation, obtaining almost 1200 hours of training data. The model was further improved with 700 hours of semi-supervised data collected in LiveATC for different airports from Europe [17]. The LM is 3-gram trained on the same data as the acoustic model with an additional textual data from additional public resources such as airlines names, airports, ICAO alphabet and way-points in Europe.

4.3. Evaluation

Since this paper focuses on improving callsign detection, we evaluate the proposed methods by calculating the accuracy of callsign extraction. For the evaluation we use ICAO format, which is the target form to display on the screen of ATCo and pilots, and we have only two outcomes: ICAO is recognized 'correctly' VS 'incorrectly'. In the previous studies [5, 6], the accuracy of callsign recognition is evaluated with matching the ground truth callsign n-grams to the ones in utterances. This approach, however, does not correspond to the real situation, when ground truth callsigns are not available. In our experiments, we do not only do speech recognition but proceed with callsign extraction, we evaluate the performance directly on the extracted entities. In addition, the use of the ICAO format helps to avoid issues with variability of pronunciation within a callsign: the full form of callsign is extracted automatically but a speaker says a shorten version, which is then outputted by the ASR, as well as recorded in the ground truth transcriptions (see example above 3.2.2). All experiments share the same ASR and BERT-based NER systems, as well as the ICAO extractor module; thus, the performances are only impacted by the proposed boosting techniques.

[1] Streaming audio platform that gathers VHF aircraft communications

[2] From the 'standard' MALORCA test sets [18] only utterances with the available surveillance information are selected.

[3] https://www.haawaii.de/wp/

[4] The ATCO2 test set is publicly available in https://www.atco2.org/data

Table 3. Results of callsign extraction with ASR boosting (ASR-B) and post-boosting (NLP-B): the accuracy of callsign recognition (%) is calculated for the callsigns in ICAO format (see Section 4.3)

Method					**Test sets (callsign recognition accuracy)**			
					LiveATC	Prague	Vienna	NATS
ASR output	Callsign extraction (baseline)				42.8	64.4	48.4	35.2
		Lattice rescoring	G-extension	NLP-boosting				
		✓	-	-	53.1	66.9	59.6	37.1
		-	✓	-	44.4	64.3	49.2	34.8
		✓	✓	-	52.8	66.9	52.1	36.8
		-	-	✓	88.4	**95.0**	**86.0**	87.0
		✓	-	✓	**88.5**	94.8	84.3	**88.9**
		-	✓	✓	87.7	**95.0**	85.6	88.2
		✓	✓	✓	88.0	94.7	84.0	88.0
Ground Truth	Callsign extraction (oracle)				**89.7**	72.2	59.6	67.4
	+ NLP-Boosting				89.3	**95.4**	**87.0**	**94.0**
ASR WER	**(without boosting)**				32.4	3.4	9.2	24.4

5. RESULTS

As a baseline we use callsign extraction done directly on the outputs of our ASR system. Then, we apply the proposed boosting techniques (G-extension, lattice rescoring, NLP-boosting) in different combinations to see how they can benefit from each other. In Table 3, the results of the experiments are presented on four different test sets with accuracy of callsign (ICAO) recognition. Overall, the proposed metrics help to improve the baseline accuracy from 30.6% to 53.7% absolutely, or from 32.1% to 60.4% relatively (for the test sets Prague and NATS correspondingly; when the NATS set gets the highest improvement being the out-of-domain data). The best results are always achieved with the use of NLP-boosting. For LiveATC and NATS sets, the out-of-domain sets in the ASR training, the best performance is achieved with the combination of NLP-boosting and ASR-boosting (lattice rescoring) methods.

At the same time, the G-extension has a contradicting effect. It helps to improve results comparing to the baseline for the LiveATC and Vienna sets, yet, its combination with lattice rescoring achieves worse accuracy than lattice rescoring alone. The possible drawback of the G-extension method is that a very high number of available callsigns are boosted in LM FST (see last column 2). It can introduce confusion when combining with the lattice rescoring boosting method, which focuses on only current callsigns. On the other hand, it does not need any modifications during the decoding and serves as a general domain adaptation. Thus, G-extension can be used to improve the outputs when other methods are not available, otherwise, can be skipped. The number of callsigns used to boost the ASR outputs may also have the degradation effect on the performance of the lattice rescoring approach. Although in this case, the number of callsigns did not exceed 50, we investigated its impact. The test sets have different numbers of boosted n-grams, from 5 to 50 (see Table 1), but even with 50 boosted callsigns the recognition accuracy goes considerably up comparing to the baseline.

Along with the evaluation of boosting methods on the ASR outputs, we provide the ‘oracle’ results, when callsigns are extracted on the ground truth transcriptions (2nd line in Table 3). This comparison allows estimating the impact of the proposed methods to the callsign extraction improvement, when no ground truth information is available. Even if the ‘oracle’ scores always stay better, the accuracy achieved with our systems shows close and comparable results. No improvement with NLP-boosting on the ground truth transcription for LiveATC test set can be explained by already high accuracy of callsign extraction, as LiveATC data was used to fine-tune the NER.

Table 4. Examples of improved callsign recognition (bold part)

Baseline (incorrect ICAO)	Boosted (correct ICAO)
wizz air four one six (**WZZ**416)	**iceair** four one six (**ICE**416)
easy three delta (**EZY**3D)	**fraction eight eight** three delta (**NJE88**3D)
serbia one nine lima (**ASL**19L)	**stobart** one nine lima (**STK**19L)

Table 4 gives examples of improvement where airline names and callsigns are detected correctly comparing to the baseline predictions. Our methods demonstrate consistent results for data of different quality. The level of noise in the recordings of LiveATC and Malorca test sets is very different, as well as WERs achieved by their baseline ASR systems (the last line in Table 3; [6]). Nevertheless, we see considerable improvement for all test sets and the general tendency stays the same. The main advantage of the proposed approach comparing to the others is its simplicity and flexibility. The NER-system can be fine-tuned to different data sets that makes it easy to adapt to new out-of-domain data. Moreover, it is also suitable for the online implementation.

6. CONCLUSION

We investigated a two-step approach of integrating contextual radar data in order to dynamically improve the recognition of callsigns per utterance. We demonstrated that the best result is achieved with the NLP-boosting and with the combination of NLP-boosting and lattice rescoring methods on all test sets of different recording quality with the significant improvement, i.e., from 32.1% to 60.4% of relative improvement on callsign recognition accuracy across the evaluated data sets. Introduction of contextual information considerably improves recognition of callsigns and, thus, recognition of ATCo messages in general. As a noisy environment leading to lower recognition accuracy is often a reality in pilot-ATCo communication, the proposed methods and their combination will definitely benefit the recognition of the key information in ATCo speech.

7. REFERENCES

[1] Anna Schmidt, Youssef Oualil, Oliver Ohneiser, Matthias Kleinert, Marc Schulder, Arif Khan, Hartmut Helmke, and Dietrich Klakow, "Context-based recognition network adaptation for improving on-line asr in air traffic control," in *2014 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2014, pp. 13–18.

[2] Todd Shore, Friedrich Faubel, Hartmut Helmke, and Dietrich Klakow, "Knowledge-based word lattice rescoring in a dynamic context," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[3] Youssef Oualil, Marc Schulder, Hartmut Helmke, Anna Schmidt, and Dietrich Klakow, "Real-time integration of dynamic context information for improving automatic speech recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[4] Youssef Oualil, Dietrich Klakow, Gyorgy Szaszák, Ajay Srinivasamurthy, Hartmut Helmke, and Petr Motlicek, "A context-aware speech recognition and understanding system for air traffic control domain," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 404–408.

[5] Martin Kocour, Karel Veselỳ, Alexander Blatt, Juan Zuluaga Gomez, Igor Szöke, Jan Cernocky, Dietrich Klakow, and Petr Motlicek, "Boosting of Contextual Information in ASR for Air-Traffic Call-Sign Recognition," in *Proc. Interspeech 2021*, 2021, pp. 3301–3305.

[6] Iuliia Nigmatulina, Rudolf Braun, Juan Zuluaga-Gomez, and Petr Motlicek, "Improving callsign recognition with air-surveillance data in air-traffic communication," Idiap Research Institute, 2021, pp. 1–5, Idiap Research Institute.

[7] Juan Zuluaga-Gomez, Iuliia Nigmatulina, Amrutha Prasad, Petr Motlicek, Karel Veselỳ, Martin Kocour, and Igor Szöke, "Contextual Semi-Supervised Learning: An Approach to Leverage Air-Surveillance and Untranscribed ATC Data in ASR Systems," in *Proc. Interspeech 2021*, 2021, pp. 3296–3300.

[8] Keith Hall, Eunjoon Cho, Cyril Allauzen, Francoise Beaufays, Noah Coccaro, Kaisuke Nakajima, Michael Riley, Brian Roark, David Rybach, and Linda Zhang, "Composition-based on-the-fly rescoring for salient n-gram biasing," 2015.

[9] Petar Aleksic, Mohammadreza Ghodsi, Assaf Michaely, Cyril Allauzen, Keith Hall, Brian Roark, David Rybach, and Pedro Moreno, "Bringing contextual information to google speech recognition," 2015.

[10] Jack Serrino, Leonid Velikovich, Petar S Aleksic, and Cyril Allauzen, "Contextual recovery of out-of-lattice named entities in automatic speech recognition.," in *Interspeech*, 2019, pp. 3830–3834.

[11] Hartmut Helmke, Matthias Kleinert, Oliver Ohneiser, Heiko Ehr, and Shruthi Shetty, "Machine learning of air traffic controller command extraction models for speech recognition applications," in *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*. IEEE, 2020, pp. 1–9.

[12] "All clear phraseology manual," in *Eurocontrol, Brussels, Belgium*, 2011, "[Online; accessed 10-September-2021]".

[13] Mehryar Mohri, Fernando Pereira, and Michael Riley, "Weighted finite-state transducers in speech recognition," *Computer Speech & Language*, vol. 16, no. 1, pp. 69–88, 2002.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[15] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, and Morgan Funtowicz et al, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 2020, pp. 38–45, Association for Computational Linguistics.

[16] Juan Zuluaga-Gomez, Karel Veselỳ, Alexander Blatt, Petr Motlicek, Dietrich Klakow, Allan Tart, Igor Szöke, Amrutha Prasad, Saeed Sarfjoo, Pavel Kolčárek, et al., "Automatic call sign detection: Matching air surveillance data with air traffic spoken communications," in *Multidisciplinary Digital Publishing Institute Proceedings*, 2020, vol. 59, p. 14.

[17] Banriskhem Khonglah, Srikanth Madikeri, Subhadeep Dey, Hervé Bourlard, Petr Motlicek, and Jayadev Billa, "Incremental semi-supervised learning for multi-genre speech recognition," in *Proceedings of ICASSP 2020*, 2020.

[18] Ajay Srinivasamurthy, Petr Motlicek, Ivan Himawan, Gyorgy Szaszak, Youssef Oualil, and Hartmut Helmke, "Semi-supervised learning with semantic knowledge extraction for improved speech recognition in air traffic control," in *Proc. of the 18th Annual Conference of the International Speech Communication Association*, 2017.

[19] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *IEEE workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011.

[20] Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahremani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur, "Purely sequence-trained neural networks for asr based on lattice-free mmi.," in *Interspeech*, 2016, pp. 2751–2755.

[21] Juan Zuluaga-Gomez, Petr Motlicek, Qingran Zhan, Karel Veselỳ, and Rudolf Braun, "Automatic Speech Recognition Benchmark for Air-Traffic Communications," in *Proc. Interspeech 2020*, 2020, pp. 2297–2301.

LAST PAGE