INTEGRATING LANGUAGE IDENTIFICATION TO IMPROVE MULTILINGUAL SPEECH RECOGNITION

Holger Caesar

Idiap-RR-24-2012

JULY 2012

Centre du Parc, Rue Marconi 19, P.O. Box 592, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

Integrating Language Identification to improve Multilingual Speech Recognition

Bachelor Thesis of Holger Caesar
Supervised by David Imseng and Prof. Hervé Bourlard
IDIAP Research Institute - Martigny, Switzerland
Evaluated by Prof. Tanja Schultz
Cognitive Systems Lab (CSL) - Karlsruhe, Germany

22 June, 2012

Abstract

The process of determining the language of a speech utterance is called Language Identification (LID). This task can be very challenging as it has to take into account various language-specific aspects, such as phonetic, phonotactic, vocabulary and grammar-related cues.

In multilingual speech recognition we try to find the most likely word sequence that corresponds to an utterance where the language is not known a priori. This is a considerably harder task compared to monolingual speech recognition and it is common to use LID to estimate the current language.

In this project we present two general approaches for LID and describe how to integrate them into multilingual speech recognizers. The first approach uses hierarchical multilayer perceptrons to estimate language posterior probabilities given the acoustics in combination with hidden Markov models. The second approach evaluates the output of a multilingual speech recognizer to determine the spoken language.

The research is applied to the MediaParl speech corpus that was recorded at the parliament of the canton of Valais, where people switch from Swiss French to Swiss German or vice versa. Our experiments show that, on that particular data set, LID can be used to significantly improve the performance of multilingual speech recognizers. We will also point out that ASR dependent LID approaches yield the best performance due to higher-level cues and that our systems perform much worse on non-native data.

Hiermit bestätige ich, dass ich diese Arbeit alleine und ohne fremde Hilfe erstellt habe. Ich habe keine anderen als die angegebenen Hilfsmittel verwendet.

I hereby confirm that this thesis was created by myself and without any foreign support. I have used no other auxiliary means than those enlisted here.

____________________ Lausanne, 22 June 2012

Contents

List of Tables

List of Figures

List of Abbreviations

AM: Acoustic Model
ANN: Artificial Neural Network
ASR: Automatic Speech Recognition
HMM: Hidden Markov Model
IPA: International Phonetic Alphabet
LCT: Linuistic Context
LID: Language Identification
LM: Language Model
MLP: Multi-Layer Perceptron
SAMPA: Speech Assessment Methods Phonetic Alphabet
VAD: Voice Activity Detection
VLID: Language Identification including Voice Activity Detection

1 Introduction

The object of this Bachelor thesis is to study the integration of Language Identification (LID) into multilingual speech recognition. LID describes the question *which* language is being spoken. In a multilingual environment this information can then be exploited in various ways to recognize *what* was said.

A possible approach for the speech recognition task is to setup one system for multiple languages. This system requires a pronunciation lexicon with words from every language and a language model that uses these words. The phoneme sets of the different languages have to be concatenated. The increased search space for a a joint language model, pronunciation lexicon and phoneme set can lead to a decrease in performance as compared to monolingual systems. To overcome this deficiency, several approaches can be applied to bias the acoustic model and the language model towards the detected language.

To recognize the language being spoken we require information about how to discriminate between languages, such as phonetic, phonotactic, vocabulary and grammar-related cues. For this task we extract acoustic features and classify them by using hierarchical Multilayer Perceptrons (MLP). In a first step we retrieve phone class posteriors. Then we use them to compute language posterior probabilities. The language posteriors from the MLPs are used as emission probabilities of a Hidden Markov Model that provides us with the correct timing. Different back-end metrics are presented and the systems are evaluated in terms of accuracy.

The LID results can then be used to choose from a set of monolingual speech recognizers or to combine monolingual phone class posteriors for a multilingual speech recognizer. We also evaluate the inverse situation and study if the result from the multilingual speech recognizer can be used to improve LID.

Furthermore we investigate code-switches in our data and analyze how they effect the performance. A code-switch is a situation where one speaker changes the language during an utterance. It is a very common phenomenon in multilingual speaker communities. There has been a lot of research on a linguistic level on the nature of code-switches as well as the reason for speakers to use them [Nilep06], but relatively few studies have addressed the role of code-switches in speech recognition.

The speech corpus used for our studies has been recorded in the bilingual canton of Valais in Switzerland. The languages spoken there are Swiss German and French. Several utterances contain code-switches. The performance of our systems is evaluated with respect to sentence duration, nativeness, channel properties and on a per-speaker level.

2 Theory

This chapter describes the theory of Automatic Speech Recognition and Language Identification.

2.1 Automatic Speech Recognition

In this section we formally define Automatic Speech Recognition (ASR) and separate it into smaller tasks. We describe how to process a recorded audio signal (Section 2.1.2) and then introduce acoustic and language modeling in Sections 2.1.3 and 2.1.4, before Section 2.1.5 describes the decoding.

2.1.1 Introduction

ASR describes the conversion of speech signals into a sequence of words W = $w_1 w_2 .. w_n$. As described by Young [Young96], this means that we search for the most probable word sequence $\hat{W}$ given the observed sequence of acoustic vectors $\mathbf{Y}$. Since we cannot compute the required probability $P(W|\mathbf{Y})$ directly, we use Bayes' rule to decompose it into its components:

$$\hat{W} = \arg\max_W P(W|\mathbf{Y}) = \arg\max_W \frac{P(W)P(\mathbf{Y}|W)}{P(\mathbf{Y})}$$

The denominator $P(\mathbf{Y})$ describes the probability of observing the acoustic signal $\mathbf{Y}$. Since $P(\mathbf{Y})$ is independent of the word sequence W, it does not effect the maximization of $P(W|\mathbf{Y})$. The expression $P(\mathbf{Y}|W)$ describes the probability of observing an acoustic signal $\mathbf{Y}$ given the word sequence W. The estimation of this probability is described in Section 2.1.3. $P(W)$ is the likelihood of observing a word sequence W. It is dependent on the language of the utterance. In Section 2.1.4 we describe how to compute this probability.

2.1.2 Preprocessing

To perform ASR, we need to preprocess the signal. An important assumption in ASR is that speech signals can be regarded as "stationary over an interval of a few milliseconds" [Young96]. That is why we can divide the signal into blocks of a fixed length and assign a feature vector to each block. The blocks typically have a length of 25ms and overlap each other by 10ms. We apply a hamming window and amplify high frequencies to "compensate for the attenuation caused by the radiation from the lips" [Young96].

For spectral analysis we do a Fast Fourier Transform (FFT) at a sampling rate of 16kHz. We use Perceptual Linear Prediction (PLP) as described in [Hermansky89] and [Robinson96]. The PLP extraction is based on a Mel-frequency filterbank. The filterbank coefficients are “weighted by an equal-loudness curve and compressed by taking the cubic root“ [HTK06]. We estimate LP coefficients and convert them to cepstral coefficients. In this study we typically extract 39 PLP features ($C_0 - C_{12} + \Delta + \Delta\Delta$).

2.1.3 Acoustic Modeling

As mentioned in 2.1.1, the purpose of the acoustic model is to find the likelihood of an acoustic signal $\mathbf{Y}$ given a word sequence W. This can be done by collecting many examples of the word W and extracting statistics of the corresponding vector sequences. However this approach is not applicable for an unconstrained vocabulary, since there is not enough data for the words and we are not able to synthesize the probabilities of unseen words. Instead we divide every word into sub-sequences of speech sounds called *phones*. *Phonemes* are an abstraction of a set of phones. They are defined as “the smallest segmental unit of sound employed to form meaningful contrasts between utterances” in [IPA99].

A problem with these monophones is that their articulation is highly dependent on the acoustic context. To achieve good phone discrimination, we model the pronunciation of a word with triphones (i.e. the phone and its left and right context).

Various approaches exist to estimate the most likely triphone sequence. The most common one is to model each triphone with a 3 state left-to-right Hidden Markov Model (HMM). An HMM is a statistical Markov model where the states are unobserved. It is defined by five components (see Section 4.1 in [Schultz06]):

- A set $S := \{S_1, S_2, ..., S_N\}$ of N HMM states
- A probability distribution π that assigns a probability to each state S_i to be the initial state q_1 of a state sequence $\pi_i = P(q_1 = S_i)$, $i = 1...N$
- A matrix $\mathbf{A} = (a_{ij})$ of state-transition probabilities, where $a_{ij} = P(q_t = S_j \mid q_{t-1} = S_i)$, $i, j = 1...N$ describes the probability to transition from state S_i to S_j
- A set of K observation symbols $V := \{v_1, v_2, ..., v_K\}$ to be emitted per time frame by the observable stochastic process
- A matrix $\mathbf{B} = (b_j(k))$ of emission probabilities, where $b_j(k) = P(o_t = v_k \mid q_t = S_j)$, $j = 1...N$, $k = 1...K$, is the probability of emitting the observation $o_t = v_k$ in state S_j

If the observations are drawn from continuous space, we use a continuous HMM. The output probability density functions $b_j(\mathbf{x})$ can for example be multivariate Gaussian mixture density functions [Schultz06]:

$$b_j(\mathbf{x}) = \sum_{l=1}^{L_j} c_{jl} \cdot N(\mathbf{x} \mid \mu_{jl}\Sigma_{jl})$$

$$\sum_{l=1}^{L_j} c_{jl} = 1$$

where L_j is the number of Gaussian mixtures, c_{jl} is the weight of mixture l in state S_j and $N(\mathbf{x} \mid \mu, \Sigma)$ denotes a single Gaussian density function with mean vector μ and covariance matrix Σ:

$$N(\mathbf{x} \mid \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \cdot e^{\frac{1}{2}(x-\mu)^T \sum_i^{-1}(x-\mu)}$$

2.1.4 Language Modeling

The language model can be trained to estimate the probability of a word w_k given its preceding words $W_1^{k-1} = w_1...w_{k-1}$. In practice this is not feasible for longer utterances. Therefore we usually compute N-grams, which means that we approximate the probability of a word sequence by its N last words [Young96]:

$$P(w_k \mid W_1^{k-1}) = P(w_k \mid W_{k-n+1}^{k-1})$$

This information is useful since it encodes “syntax, semantics and pragmatics” [Young96] on a local level.

However, due to the complexity of the number of possible N-grams (V^N for V words in the vocabulary), there is a data sparsity problem. We can overcome this by using *discounting* and *back-off*. Discounting means that we decrease the probability of more frequently occurring N-grams and redistribute it to less frequent N-grams. Back-off is applied when there is not enough data to train an N-gram model. In that case we replace the N-gram probability by it's $(N-1)$-gram approximation and a scaling factor.

This approach does not take into account grammatical rules on a sentence level. Various approaches have been suggested to overcome this deficiency. However, N-grams give a good trade-off between computational costs and accuracy and are therefore today prevalent in ASR.

2.1.5 Decoding

The *Decoding Problem* can be described as follows (see [Schultz06]):
"What is the state sequence q^* that most likely generated the observation sequence $O = o_1 o_2 ... o_T$ for a given HMM $\lambda = (A, B, \pi)$?"
Depth-first and breadth-first approaches can be used to solve this search problem [Young96]. Typically a breadth-first algorithm is used which is called *Viterbi decoding*. Due to the combinatorial explosion of the search space, this problem cannot be solved on a large vocabulary in reasonable time. Therefore we apply pruning based on beam search. This means that at any point in time during the decoding, we reject all hypotheses that have a lower probability than the solution subtracted by a threshold.

2.2 Language Identification

Language Identification (LID) is the task of recognizing the language of an utterance. In LID we reduce the complexity of a given utterance to lower dimensional information such as a probability for any possible language. Since there are about 6,900 different spoken languages [SIL92], we can only consider a subset.

The differentiation between a language and a dialect is not clearly defined and often subject to political rather than linguistic considerations (e.g. the term Swiss German refers to a group of Alemannic dialects spoken in Switzerland). Differences between languages can exist on all linguistic levels (such as different words, grammatical structures, phoneme sets, acoustic realization, phonotactic constraints, prosody, grapheme to phoneme relation etc.). Different approaches for LID have been proposed in [Zissman01]:

- **Spectral-similarity approaches:**
  In these approaches several short-term spectra are extracted from the speech utterances. The spectra of the test utterances are then compared to those of the training utterances, using an Euclidean, Mahalanobis or another distance metric. The distant scores are accumulated and the language with the lowest distance score is selected.

- **Prosody-based approaches:**
  These approaches are based on pitch estimation and amplitude contours. They are then normalized to be “insensitive to overall amplitude, pitch and speaking rate” [Zissman01]. The accuracy of prosody-based approaches is highly language pair specific.

- **Phone-recognition approaches:**
  Phone-recognition approaches investigate the phone inventory of an utterance. Language characteristics are extracted based on the temporal order of the phones. Phonotactic constraints can be use in N-gram analysis to improve the result. These approaches require phonetically-labeled corpora, but typically yield a higher performance. We use a phone-recognition approach in Section 4.2.

To do LID, we follow the approach presented by [Imseng10]. It is based on the use of Artificial Neural Networks which are introduced in Section 2.2.1. A more detailed description of the approach is given in Section 4.2.

2.2.1 Artificial Neural Networks

The term Artificial Neural Network (ANN) describes a mathematical model that is widely used in machine learning. The model is inspired by biological neural networks such as the human brain. An ANN is a connectionist approach to approximate mathematical functions with a finite number of input and output dimensions. It is used to detect patterns that are not known in advance. This can be seen as a contrast to expert systems that rely on rules predefined by the knowledge worker.

An ANN is a network of units ("neurons") that perform a non-linear function on their inputs. This is typically a scalar function of the weighted sum of the inputs [Bengio96]:

$$y_i = f\left(\sum_j w_{ij} y_j + b_i\right)$$

with y_i being the average firing rate of a unit i, b_i the bias or threshold of a unit i and w_{ij} being the weight between units i and j. The weights and biases are free parameters that are trained as described below. The activation function f is typically a non-linear squashing function.

Various forms of Artificial Neural Networks exist such as Feed-forward Neural Networks, Time Delay Neural Networks and Recurrent Neural Networks. Due to the approach followed in this report, we limit our focus to Feed-forward Neural Networks, more specifically Multilayer Perceptrons (MLPs). MLPs are ANNs where each layer is fully connected to the next one. Except for the input nodes, each node is a neuron with nonlinear activation function. The activation function of a MLP is typically a logistic function (1) or a hyperbolic tangent (2) (see [Bengio96]):

$$f(x) = \frac{1}{1+\exp(-x)} \text{ (1)}$$

$$f(x) = \tanh x \text{ (2)}$$

The advantage of the MLP architecture is that it can be trained very efficiently. The *Universal Approximation* theorem states that a MLP can approximate any continuous function with arbitrary precision, given enough hidden units [Cybenko89].

Training Procedure We use a supervised training method called *backpropagation* to gradually reduce a differentiable cost function. Hence we approach a local minimum of the cost function by iteratively applying a discrete gradient descent algorithm. As a cost function we typically use the mean square error criterion, which is the square of the differences of activation of output units $y_{p,i}$ and their target values $target_{p,i}$ for unit i and pattern p [Bengio96]:

$$C_{MSE} = \sum_p C_p = \sum_p \sum_i (y_{p,i} - target_{p,i})^2$$

To optimize the aforementioned criterion we iteratively apply a deterministic weight update over all training patterns p [Bengio96]:

$$\theta \leftarrow \theta - \epsilon \sum_p \frac{\delta C_p}{\delta \theta}$$

where C_p is the local cost of pattern p, ϵ the learning rate and θ the parameter of the network. The learning rate has to be carefully chosen to allow convergence to a local minimum. Research shows that it has to satisfy the following conditions [Bengio96]:

$$\sum_{t=1}^{\infty} \epsilon_t = \infty$$

$$\sum_{t=1}^{\infty} \epsilon_t^2 < \infty$$

We use the following strategy to train the MLPs in this report: We begin our training with a learning rate of ϵ_0 (e.g. $\epsilon_0 = 0.0008$). Successive training steps are applied until the performance on the development set deteriorates. Then we continue training with $\epsilon_t = \frac{\epsilon_t}{2}$. This way we allow for efficient convergence.

3 The MediaParl Speech Corpus

3.1 Description

The MediaParl speech corpus was recorded at the cantonal parliament of Valais, Switzerland. It consists of about 20 hours of French recordings and 20 hours of Swiss-German recordings. After neglecting sentences of bad quality there are a total of 15,576 utterances.

The recordings took place in 2006 and 2009 with sampling rates of 16kHz or 44.1kHz. In the latter case, recordings are down-sampled to 16kHz. The speech recordings are manually transcribed and sentence boundaries are labeled.

We differentiate between two different data sets. The sentences data set is made up of all sentences in MediaParl. The interventions data set is just a subset of the test data in sentences. In contrast to those, interventions are not split on a sentence level and include 18 code-switched and 18 mono-lingual utterances.

The term code-switch refers to a change of languages during an utterance. The change can occur on an inter-sentential or an intra-sententional level. Since the latter does not exist in MediaParl, we define code-switches as language switches at the sentence boundary.

The annotation statistics of the speech corpus are shown in 1. It reveals that French sentences contain a higher number of words with a shorter duration. It can also be seen that German sentences tend to be shorter as compared to French sentences. Given the goal of this study, the speaker data is divided as follows: 90 % of the monolingual speakers (83.2 % of total)

Table 1: Annotation Statistics on MediaParl

	French	**German**	**Total**
Total sentences annotated (in hours)	20:34:31	20:02:06	40:36:37
Number of sentences	7,058	8,530	15,588
Number of words	203,614	159,873	363,487
Sentences per speaker (mean)	56	97	76
Sentences per speaker (median)	25	52	30
Sentences per speaker (min)	2	5	2
Sentences per speaker (max)	511	698	720
Words per second	2.749	2.217	2.486
Sentences per second	0.095	0.118	0.107
Words per sentence	28.85	18.74	23.32

are used as training set and the remaining 10 % as development set. The bilingual speakers (16.8 % of total) form the test set.

We choose this approach to explore if a system trained on monolingual speakers can be successfully applied to bilingual speakers and data containing code-switches. In general it might be better to train the system by using only bilingual speech data, but due to a lack of data this is not feasible on the MediaParl speech corpus. Note that only 18 recorded interventions (691 out of 15,588 sentences) include code-switches and only 6 speakers switch the language within an intervention. 11 out of 18 interventions are spoken by the same speaker that seems to be very fluent in both languages.

3.2 Creating the Dictionaries

To align speech recordings with the respective transcription, we need a dictionary for each language. The dictionary maps the grapheme representation of a word to its pronunciation.

The phonemes in our dictionaries are represented using the Speech Assessment Methods Phonetic Alphabet (SAMPA) [1]. SAMPA is based on the International Phonetic Alphabet (IPA), but features only ASCII characters. It was developed under the ESPRIT, BABEL and COCOSDA projects and supports multiple languages including German and French.

The creation of a dictionary can be very time-consuming. Manual processing requires a language expert to expand the words from transcription into their pronunciation. Therefore we use dictionaries that are already publicly available, but are designed for a more general domain of speech, such as conversations. The speech corpus that we use includes large amounts of words, that are specific to the domain (politics) and region (Switzerland). Hence, the dictionary needs to be completed.

As a starting point we use the grapheme-to-phoneme (g2p) framework Phonetisaurus. G2p frameworks use existing dictionaries to define common rules that map a sequence of letters in the written representation (grapheme) to their acoustic representation (phoneme). The rules are then applied to unseen words. The g2p approach has certain limits. A study on CMUDict yielded a word error rate of 24.4 % [2]. Languages such as English have limited grapheme to phoneme correspondence. Therefore it is relatively difficult to derive simple mapping rules from the grapheme representation of a syllable to its phoneme representation.

The English words “how” and “out”, for example are spelled very differently, but use the same diphone /aU/. There are also heteronyms that have the same spelling, but a different pronunciation based on their grammatical function. The word “insult” has a different stress on its syllables depending on whether it is used as a verb (/In”sVlt/) or a noun (/”In.sVlt/).

Another problem is that the rules of one language do not necessarily generalize to a foreign language. As an example we can look at word endings in French and German. In the case of the French language, word suffixes are often not pronounced if they form an extension to the word stem, such as plurals and conjugations (i.e. adultes, /a d y l t/). In contrast to that, in the (Swiss) German language word suffixes are usually pronounced, but in some cases terminal devoicing (“Auslautverhärtung”) applies. Terminal devoicing means that voiced consonants become unvoiced before vowels or breaks (Süd

[1] http://www.phon.ucl.ac.uk/home/sampa/index.html

[2] http://code.google.com/p/phonetisaurus/

/zy:t/, Süden /zy:d@n/).

Also note that even if we have a model that can sufficiently describe the phenomena of one language, we also include frequently appearing words of a foreign language into our dictionary. Due to the prevalence of the English language in many fields, domain-specific words are often borrowed from English. Example domains from our dictionary include technology ("computer", "highspeed"), media ("interview") and business economics ("controlling"). Since MediaParl was recorded in a bilingual canton, this problem becomes even more important than in other more homogenous speaker populations.

Due to these problems with g2p, all entries generated by Phonetisaurus were manually verified by native speakers according to the SAMPA rules for the respective language. Furthermore we tried to be coherent with the Phonolex dictionary.

This approach was chosen due to pragmatic considerations and the lack of a suitable Swiss German dictionary or the rules on how to create it using SAMPA notation. Systematic discrepancies between German and Swiss German can therefore not be ruled out and have to be taken into account during evaluation. Further linguistic research on this topic might therefore be useful.

Table 2 shows the number of unique words in each dictionary.

Table 2: Vocabulary size

Language	**#words**
German	16,778
French	12,362
Bilingual	29,140

3.2.1 Swiss German Dictionary

To create the Swiss German dictionary we used the pronunciation lexicon Phonolex. Phonolex was developed by a cooperation between DFKI Saarbrücken, the Computational Linguistics Lab, the Universität Leipzig (UL) and the Bavarian Archive for Speech Signals (BAS) in Munich [3]. The pronunciation is coded in extended SAMPA.

82.0 % of the unique German words in MediaParl were found in Phonolex. Phonetisaurus was trained on Phonolex and then used to generate pronunciations for the remaining 3060 words. Table 3 shows examples of foreign words that were incorrect after doing the g2p conversion: All g2p entries were manually corrected in accordance to the German SAMPA rules in [4]. Some rules might only apply to German. For example, it suggests to use a phoneme /Q/ (or /?/ in their notation) to model the glottal stop before a vowel. We differentiate between open and closed, as well as checked and free vowels. To be able to model French loan words, we introduce nasals with the same possible modifications. We furthermore allow free diphthongs (/aI/, /aU/, /OY/), but split affricates (/p f/, /t s/, /t S/, /d Z/). Note that throughout this report, the absence of phonemes, which is silence, is also referred to as a phoneme.

Table 3: Examples of German g2p conversion failing

Word	**g2p**	**Pronunciation**
quand	/k v a n t/	/k a~:/
boillat	/b o i: l l a: t/	not possible without /w/
politique	/p o: l i t i: k v @/	/p o: l i t i: k/

As mentioned before, for some words the standard German pronunciation differs significantly from the Swiss German pronunciation. A comparison of different pronunciations of the German word “achtzig” reveals that speakers in MediaParl pronounce this word in 3 different ways. See Appendix 7.1 for our list of phonemes:

1. /Q a x t s I C/
2. /Q a x t s I k/
3. /Q a x t s I k C/

[3] http://www.bas.uni-muenchen.de/forschung/Bas/BasPHONOLEXeng.html

[4] http://www.phon.ucl.ac.uk/home/sampa/

1) is the Standard German version of the word that can be found in Phonolex and 2) can be found in various German dialects. In contrast, 3) seems to be a Swiss German peculiarity. From now on we will refer to Swiss German as German.

3.2.2 French Dictionary

The French dictionary includes the BDLex pronunciation lexicon[5] which uses extended SAMPA as well. We run a g2p conversion and use the French SAMPA rules. Similar to the German dictionary, several words cannot be converted. Examples are given in 4:

Table 4: Examples of French g2p conversion failing

Word	**g2p**	**Pronunciation**
bodenmüller	/b O d e m y l e R/	/b o d @ n m y l 6/
führungsreserve	/f y R u N s R e s E R v/	/f y R u N s R e s E R v @/
matterhorn	/m a t E R O R n/	not possible without /h/
weisungsbefugnis	/v a i s u N s b e f y g n i s/	/v aI s u N s b e f u g n i s/
wirklichkeit	/v i R k l i S k a j t/	not possible without /C/, /aI/

There are noticeable acoustic variations compared to the standard French pronunciation, which can be attributed to the Valaisan accent. For example, speakers tend to pronounce the closed vowel /o/ as the open vowel /O/ in words like “eau” (French for “water”).

[5] `http://www.irit.fr/~Martine.deCalmes/IHMPT/ress_ling.v1/rbdlex_en.php`

3.3 Language Model

Based on the training data we create a statistical bi-gram language model for each language. We also create a bilingual language model. Since there are no multilingual sentences, the bilingual language model can be seen as a conjunction of the monolingual language models. Words that exist in both languages are considered as the same word with multiple pronunciation variants.

A more sophisticated approach was presented in [FuegenSchultz03]. To overcome the problem of unbalanced multilingual LMs due to different quantities of training data they suggest to "balance the probability distribution functions of the two languages by assigning similar probabilities to two n-grams obtained from different corpora if they had a similar frequency rank with respect to the rest of the n-grams obtained from the respective corpus". Since the amount of French and German test data is well balanced in MediaParl, we do not apply this technique. Furthermore, Fügen et al. show that combining monolingual LMs at a meta level leads to better results.

To measure the quality of our language models we compute the perplexity on our test data as described in [Rosenfeld97]. For a distribution function $P_T(x)$ of the text T and a probability function $P_M(x)$ of the model, we compute the *cross-entropy* or *logprob* $H(P_T; P_M)$:

$$H(P_T; P_M) = -\sum_x P_T(x) * \log P_M(x)$$

The perplexity $PP_M(T)$ of T is then defined as:

$$PP_M(T) = 2^{H(P_T;P_M)}$$

Table 5 shows the perplexity of each language model on the test data. The difference in vocabulary size (see Table 2) limits the comparability between different LMs. However we can compare the perplexity of the bilingual LM on different data sets. We see that the perplexity on German data is much higher than on French data. This increase in perplexity will be important for the evaluation of the bilingual ASR systems in Section 4.3.2.

Table 5: Language model perplexity

	German Data	**French Data**	**Both**
German LM	369.1	-	-
French LM	-	153.8	-
Bilingual LM	717.2	248.7	442.5

4 Systems

This chapter describes the systems that we devise for our experiments. As explained earlier in Chapter 3.1, the MediaParl database provides us with word transcriptions. However, phoneme transcriptions are required for the studies. Manual transcriptions would be too costly, therefore, Section 4.1 first describes how we automatically transcribe (forced align) the recordings. Then, in Section 4.2 we introduce different language identification approaches and Section 4.3 describes monolingual and multilingual Automatic Speech Recognition systems.

4.1 Forced Alignment

We use various software tools including the Hidden Markov Model Toolkit [6] (HTK) and the IDIAP Speech Scripts to perform forced alignment. We used the standard procedure as given in the HTK tutorial [HTK06].

- **Feature extraction**
  We first extract Perceptual Linear Prediction (PLP) vectors from the recorded audio files. We use a hamming window of 25 ms size and an overlap of 10 ms. After a fast Fourier transform we compute the power spectrum, rescale the critical band to the Bark scale and apply a preemphasize coefficient of 0.97 to normalize the energy. The filterbank has 24 channels and the frame period between two feature vectors is 10 ms. The dimensionality of the features is 39 ($C_0 - C_{12} + \Delta + \Delta\Delta$).

- **Monophone training**
  For each phoneme including silence, a single Gaussian monophone HMMs is initialized with a 3-state left-to-right HMM prototype without skips. The mean and variance for all prototypes is set to the global mean and variance of the training data. To re-estimate the monophone HMMs, we iteratively run the the forward-backward algorithm. We also use adaptive pruning to limit the decoding time.

- **Fix silence**
  To make our silence model more robust and to allow to "absorb various impulsive noises", we refine it by adding transitions between states 2 and 4 and vice versa [HTK06]. Furthermore, We introduce a 1-state model for short pauses (sp). Short pauses may occur between words in the transcription. The peculiarity of this approach is that the one

[6] `http://htk.eng.cam.ac.uk`

state of a short pause is tied to the center state of the silence model as illustrated in Figure 1.

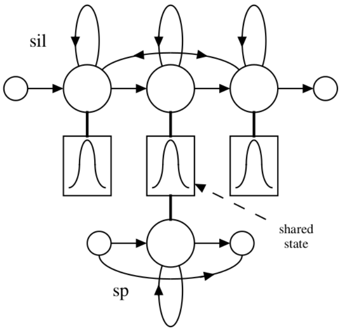


Figure 1: Short pause and silence in the Tee model

- **Triphone training**
  As usually done, we extend the context-independent monophone models to context-dependent triphone models. Each non-silence monophone is extended with its left and right context. For example the sentence “Merci.” (1) is transformed from monophone representation (2) into triphone representation (3) as given below.

  “sil merci sil”(1)
  “sil m E 6 z i: sil”(2)
  “sil sil-m+E m-E+6 E-6+z 6-z+i: z-i:+sil sil”(3)

  Similar to the monophone models, the triphone models are then iteratively re-estimated with the forward-backward algorithm.

- **State tying**
  Since there might be insufficient data to properly train some triphones, we apply state tying. As usually done, we use a decision tree that is based on context questions to cluster states [HTK06].

  The standard decision tree approach is based on the linguistic information of how phones are articulated in the human voice system. It is justified by the assumption that phones that are created in a similar manner also share a similar feature vector representation. The 428 questions include questions about general classification (labial, nasal, vowel, consonant), position of articulation (front, central, back, uvular, bilabial), voicing, roundness of the lips, airstream (fricative, approximant) and position of a phone in the triphone (front, central, back).

The root node contains the set of all triphones and then divides it into two subsets by applying the question that maximize the increase in log-likelihood. The tree is developed until either all questions have been used or the log-likelihood increase is lower than a predefined threshold. Any pair of states that can be merged without decreasing the log-likelihood more than the predefined threshold will then be tied together. The decision tree approach is also able to synthesize unseen triphones during decoding.

It is important to build such a tree for both languages independently. Two phonemes can be very different in one language, but might be close or even undifferentiable in the another language.

4.2 Language Identification

In this section we describe five different system that identify the language of an utterance. The systems presented in 4.2.2 use an hierarchical MLP approach. We furthermore present two ASR-dependent systems in 4.2.2. We then present three different methods to determine the language and compare the performance of our systems in 4.2.3.

The hierarchical LID approach makes use of two MLPs. Originally, the first layer was proposed to be a “universal phone set MLP classifier” [Imseng10]. In this study, we will also explore monolingual phone set MLP classifiers in the first layer. The resulting phoneme posterior probabilities are then used as input for the second MLP that implicitly exploits “different types of patterns/information such as confusion between phonemes and/or phonotactics for LID” [Imseng10].

4.2.1 Phoneme Classification

The first MLP (”phone MLP”) is trained to estimate phone class posteriors $P(c_t^k|x_t)$ given the acoustics x_t, where c_t^k stands for phone class $k = 1...K$ (with K being the total number of phonemes). As usually done, we consider a temporal context of 4 frames on both sides, hence in total 9 frames of 10ms each. This 90ms interval should be sufficiently long to capture single phonemes. Each frame consists of 39 PLP features and therefore our MLPs have 351 input units.

To perform LID, we study three different phone MLPs. One MLP is trained to estimate German phone posteriors, one to estimate French phone posteriors and the third one is based on the shared phonemes from both languages. To determine the shared phoneme set, we assume that the French phoneme /a/ is the same as the German phoneme /a/ and also consider phonemes like /w/ and /x/ that only appear in one language. This has the advantage of providing more training data to the shared phonemes, but on the other hand it also increases the variance in the training data of the shared phonemes, which might lead to worse LID accuracy.

The number of free parameters of all MLPs is fixed to 10 % of the number of frames available in the training data. The number of hidden units is computed accordingly and given in Table 6. The MLPs are trained with Quicknet[7] as described in Section 2.2.1.

[7] QuickNet software from `http://www.icsi.berkeley.edu/Speech/qn.html`

Table 6: Multi-layer Perceptron statistics for phone MLPs

	Units			
MLP	**Input**	**Hidden**	**Output**	**Free parameters**
German	351	1265	59	0.5M
French	351	1491	38	0.6M
Shared	351	2654	63	1.1M

4.2.2 Language Classification

The ground truth that provides the language of each sentence is required to train an LID system. For the MediaParl speech corpus the transcription already includes this information. The language ground truth of the interventions was manually created by listening to the recordings.

Hierarchical MLP Approaches

Approach 1: Shared phone LID The approach is described as "System Hier" in [Imseng10]. The first MLP is the shared phone MLP described above. The second MLP is used to classify the language, French or German, based on the shared phone class posteriors. Hence the phone class posteriors serve as features. These features are more discriminant compared to standard acoustic features such as PLPs. Therefore we can expand the temporal context to 29 frames of 10ms each. The parameters of the second MLP can be seen in Table 7.

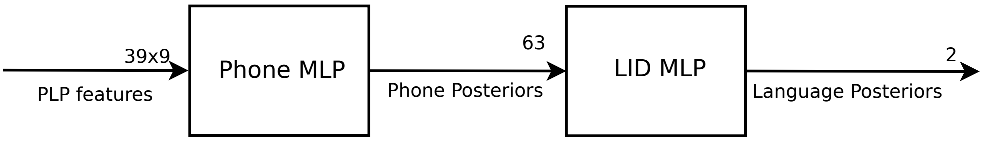


Figure 2: Shared phone LID

Approach 2: Separate phone LID In this approach we train one MLP for each language independently (separate phone MLPs). The output of these MLPs is merged, resulting in a vector with 97 dimensions. In this case, the German and French phoneme /a/ are regarded as different phone classes. The merged vectors serve as input to the second MLP that classifies the language. The number of units in each layer of this MLP can be found in Table 7.

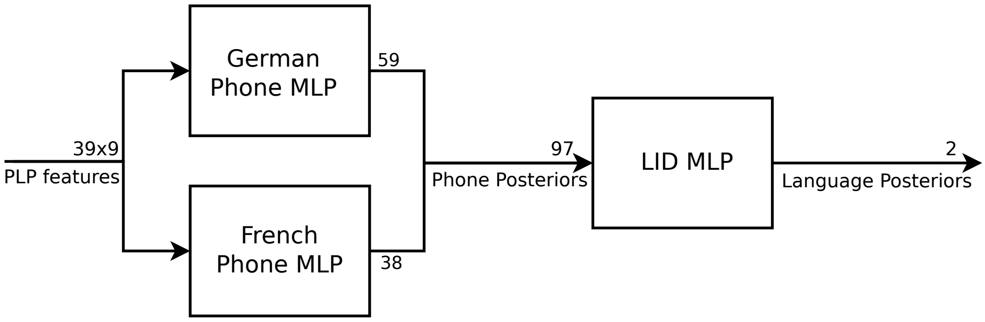


Figure 3: Separate phone LID

Approach 3: Voice Activity Detection LID The ”channel” of all data may be considered to be same, since the recording were all done in the same room and microphone setup was consistent. Classifying silence as one of the languages (in the ground truth) may be detrimental to the MLP training. Furthermore, we assume that VAD can improve the detection of code-switches since they appear at sentence boundaries which usually contain some silence. Therefore, we introduce Voice Activity Detection (VAD) and refer to that system as VLID.

We used a two-dimensional posterior vector to perform VAD. More specifically, we use the posterior of the “phoneme” sil as a silence posterior and sum all the other phone posterior probabilities to a “speech” posterior. To obtain a smoothed sequence of speech and silence, we use a relatively unconstrained 5-state HMM that allows any combination of speech and silence. We then post-process the HMM output and replace speech tags with the respective language to create the targets for the LID MLP.

As already seen for Approach 1, we then train the LID MLP, but with 3 labels (silence, German and French) instead of a two labels (German and French).

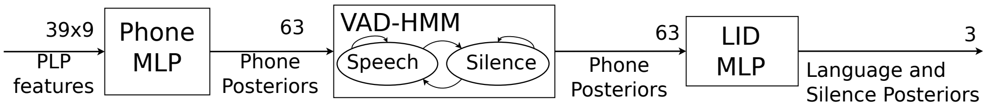


Figure 4: Voice Activity Detection LID

Table 7: Multi-layer Perceptron statistics for LID MLPs

	Units			
MLP	**Input**	**Hidden**	**Output**	**Free parameters**
Shared LID	1827	602	2	1.1M
Separate LID	2813	391	2	1.1M
Shared VLID	1827	602	3	1.1M

ASR-dependent Approaches In this section we present two LID approaches that are based on ASR as described in Chapter 4.3. Schultz et al have shown that the integration of "lexical and linguistic knowledge" leads to a reduction of LID errors of up to 50 %.

Approach 1: Shared Recognizer LID In this approach we use a joint ASR system based on shared phone class posteriors. We then analyze the ASR output and count the number of words in each of the two languages. The language with the higher word count is used as the result for LID. In contrast to the hierarchical MLP approaches, this system can also benefit from the language model of the recognizer. We use the standard shared phone class posteriors from System 1 and the linearly combined posteriors from System 4 (see 4.3).

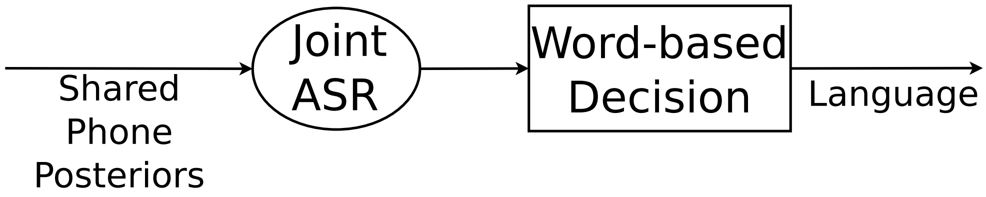


Figure 5: Shared Recognizer LID

Approach 2: Separate Recognizer LID This system is based on two monolingual recognizers. Each recognizer outputs the probabilities (normalized over time) of the most likely word at a given time. We then simply multiply the probabilities of all words. The recognizer with the higher probability determines the language. Note that this system is very similar to System 3 (see 4.3).

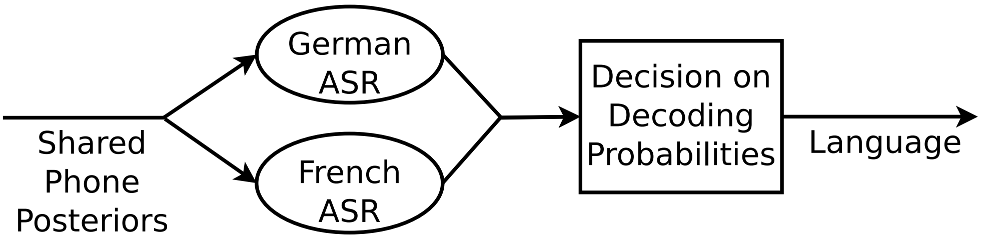


Figure 6: Separate Recognizer LID

4.2.3 Back-end Metrics

Multiplication The ASR independent approaches presented in the previous section both output a probability for every language of each frame. To evaluate the more likely language on a sentence level, we multiply the language posterior probabilities of a language $\mathcal{L}$ for each of the T frames by taking the sum of logs:

$$P(\mathcal{L}) = \prod_{t=0}^{T} p_t(\mathcal{L}) = \sum_{t=0}^{T} logp_t(\mathcal{L})$$

This approach assumes that the language posteriors are independent for each frame.

Moving Average In this approach we apply a symmetric moving average. A moving average is a finite impulse response filter that averages over N elements. This means that for language $\mathcal{L}$ and frame t we average over $d = \lfloor N/2 \rfloor$ elements on both sides of the current element:

$$p_t(\mathcal{L}) = \frac{1}{N} \sum_{n=t-d}^{t+d} p_t(\mathcal{L})$$

In contrast to Metric 1, this approach gives us not just the accumulated probabilities for each sentence, but a per frame probability. This is useful to take a closer look at code-switches.

Hidden Markov Model We now use the results from our LID MLP as input for a Hidden Markov Model (HMM). The idea of this approach is to combine the advantages of HMMs and MLPs. MLPs provide very accurate results for static patterns, whereas HMMs perform very good on time-dependent dynamic patterns, but rely on a priori knowledge. The language probabilities of the LID-MLP serve as emission probabilites for the HMM.

Our HMM has two categories of states. N internal states represent the German language and N internal states represent the French language. Each state is connected to itself and to one successor in the same language, creating a circle structure. Only one state of each language holds a transition to the other language. This topology ensures that the HMM stays in the same language for at least a certain period of time (determined by the number N of states), before it can make a transition to the other language or continue circling. See Figure 7 for an example of the topology ("Interventions in LID").

This approach has a similar effect to the "multilingual Meta-LM" presented in [FuegenSchultz03]. Whereas our approach makes a hard choice on the current linuistic context (LCT), they suggest to alter the language model during decoding based on the LCT.

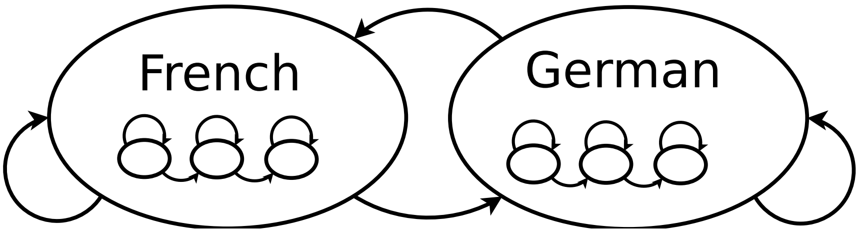


Figure 7: Hidden Markov Model for LID

To limit the number of possible decoding combinations, we define the following context-sensitive grammars as language models:

- Sentences in LID: (*german* | *french*)
  This means that a sentence is either German or French.

- Sentences in VLID: ([*silence*] *german* | *french* [*silence*])
  This means that a sentence is either German or French, but it can have optional silence at the beginning or end.

- Interventions in LID: (<*german* | *french*>)
  This means that any combination of German or French is allowed.

- Interventions in VLID: (*silence* <(*german silence*) | (*french silence*)>)
  This grammar forces a silence tag a the beginning and after each language tag.

We apply this metric on Approach 1 and 3. In the latter case we modify the topology to also include a state for silence (with several internal states).

4.3 Automatic Speech Recognition

This section describes different systems for Automatic Speech Recognition (ASR). We first introduce a monolingual ASR systems. Then we present three different approaches for bilingual ASR.

4.3.1 Monolingual ASR

We set up a monolingual ASR system for each language. As described in Section 2.1 we need an acoustic model, a pronunciation dictionary and a language model. The acoustic model is the standard setup described in 4.1 with a 3-state left-to-right HMM for each phoneme and a tee model for silence and short pauses. The emission probabilities for the HMMs are provided by the phone class posteriors from the phone MLPs in 4.2.1. The pronunciation dictionary was created as described in 3.2 and we use the monolingual statistical bigram language models from Section3.3.

4.3.2 Bilingual ASR

This section presents different systems for multilingual speech recognition. Systems 2 and 4 use the LID components developed in 4.2 and Systems 1 and 3 use implicit LID information derived by the decoder.

System 1 System 1 is an HMM system trained on the shared phone MLP phone class posteriors from Section 4.2.1. It uses a joint German and French dictionary and a statistical bigram language model as described in 3.3. This system might perform better on intra-sentential code-switches than the other systems. However, since code-switches only occur at sentence borders in MediaParl, the accuracy may decrease due to the fact that there are more words in the bilingual dictionary. The higher perplexity of the multilingual system (see Section 3.3) may lead to a higher confusion rate of words.

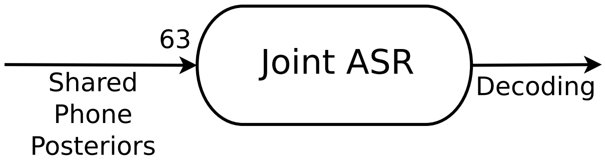


Figure 8: ASR System 1

System 2 This system uses the LID results from 4.2.2 as an input. Based on the language decision, the respective monolingual ASR system is used. Since the system is fully dependent on components trained in Sections 4.2.2 and 4.3.1, we do not do any further training. The advantage of this approach is that we can combine the robust LID performance with the specialized ASR system. The disadvantage is that the errors done in LID cannot be recovered. In the special case of 100 % LID accuracy we get the following oracle LID accuracy:

$$P_{oracle} = \frac{D(f)*P_f(x)+D(g)*P_g(x)}{D(g)+D(f)}$$

Where x is a data set, $P_{\mathcal{L}}(x)$ the monolingual performance, $\mathcal{L}$ the language and $D_{\mathcal{L}}$ the duration of the recording.

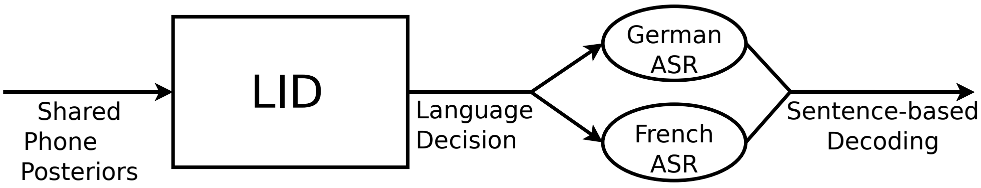


Figure 9: ASR System 2

System 3 In this approach we run both monolingual ASR systems and choose the output with a highest probability provided by the decoder. For that purpose we multiply the probabilities, normalized according to the duration, that the decoder assigns to each word given the current context. The system is more flexible than Systems 1 and 2. since the monolingual components do not depend on each other. In contrast to it, this system uses acoustic and phonotactic evidence for LID.

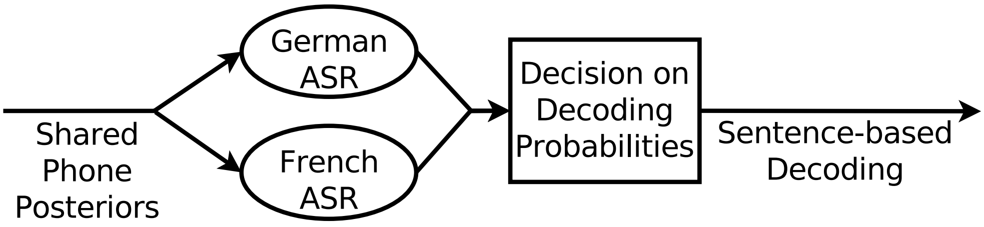


Figure 10: ASR System 3

System 4 System 4 linearly combines the phone class probabilities of different languages based on the language posteriors of the respective language. These joint phone class posteriors are then used to train a joint ASR system as in System 1.

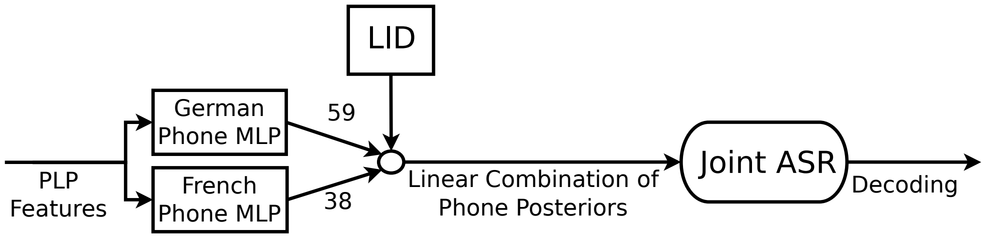


Figure 11: ASR System 4

4.4 Comparison to other Approaches

Various approaches for LID, ASR and multilingual ASR can be found in the respective literature.

In [Kumar10] the authors present a technique to vary the acoustic resolution of a phone decoder in LID by selecting the optimum set of phones. They show that a phone mapping using SAMPA/IPA as is done in this work is not neccessarily the optimum mapping for LID.

[VuKrausIS11] and [VuKraus11] describe how to build a Vietnamese and a Czech ASR system from scratch without any transcribed audio data. They use cross-language transfer from other languages, unsupervised training based on the "multilingual A-stabil" confidence score and bootstrapping. This approach is especially appropriate for under-ressourced languages. Since we are not working with under-ressourced languages in this work, different approaches were chosen.

Speech adaptation for non-native speech is a common way to improve ASR performance. Typical acoustic model adaptation techniques are Maximum Likelihood Linear Regression (MLLR) and Maximum A Posteriori (MAP) adaptation. [WangSchultz03] present a technique to improve non-native spontaneous speech recognition by using polyphone decision tree specialization. It can be assumed that such techniques would further improve our results, but they are out of the focus of this work.

5 Experiments and Discussion

This chapter presents the training and testing procedures of the systems presented above. It furthermore discusses the experimental results.

5.1 Language Identification

5.1.1 Phoneme Classification

Appendices 19 and 20 give an overview over the training procedure for the phone MLPs. About 70 % of the phones in the development set are classified correctly. Even though we have more phonemes in German, the accuracy of German and French phoneme detection yields a similar performance. Aspects that deteriorate the recognition performance include the use of foreign words, accented speech, mumbling and general noise.

Appendix 21 shows the training process of the shared phone MLP. The results are 3 % to 4 % worse on the training and 3 % to 6 % worse on the development set than for the separate phone MLPs. This supports the claim of linguists that phonemes, even though they are represented by the same symbol in IPA, might have a slightly different pronunciation in different languages.

These numbers seem to be relatively low, but note that we report frame-based accuracies. Furthermore, the actual pronunciation can be influenced by language, dialect, stress, mood and domain, as well as many other factors. If a phone has acoustic similarity to different phone classes (such as /e/, /e:/, /E/, /E:/ and /@/), it is very hard to differentiate those, even for a native speaker of the language.

5.1.2 Language Classification

Hierarchical MLP Approaches In Section 4.2.2 we introduced 3 hierarchical MLP approaches. In this section we present the training of the second MLP.

Approach 1: Shared phone LID Our shared phone LID approach yields a frame-based accuracy of 99.1 % on the training set and 97.2 % on the development set. This accuracy is measured on a per-frame base. The results of the training procedure can be seen in Appendix 22.

Approach 2: Separate phone LID The separate phone LID approach yields a frame-based accuracy of 99.2 % on the training set and 97.0 % on the development set. The results can be seen in Appendix 23. Since frame-based accuracies only differ very little from Approach 1, it is hard to decide which approach performs better.

Approach 3: Voice Activity Detection LID Our HMM results in silence (15.5 %) and speech (84.5 %). For scoring on a per-frame base, we also include the performance of silence detection into our statistics. The result of 99.0 % on the training data and 97.2 % on the development data shows that we do not lose performance by including silence, but that we gain information. This information can be useful to find a segmentation into sentences which can be exploited to detect code-switches. The results are shown in Appendix 24.

ASR-dependent Approaches As we will see later, several HMM parameters need to be tuned to get the best ASR performance.

5.1.3 Back-end Metrics

Multiplication Table 8 shows the LID results using the multiplication metric. The results are on a per-sentence base. A statistical significance test [BisaNey04] with 99 % confidence shows that the LID results are not significantly different.

Table 8: Language Identification results with multiplication metric

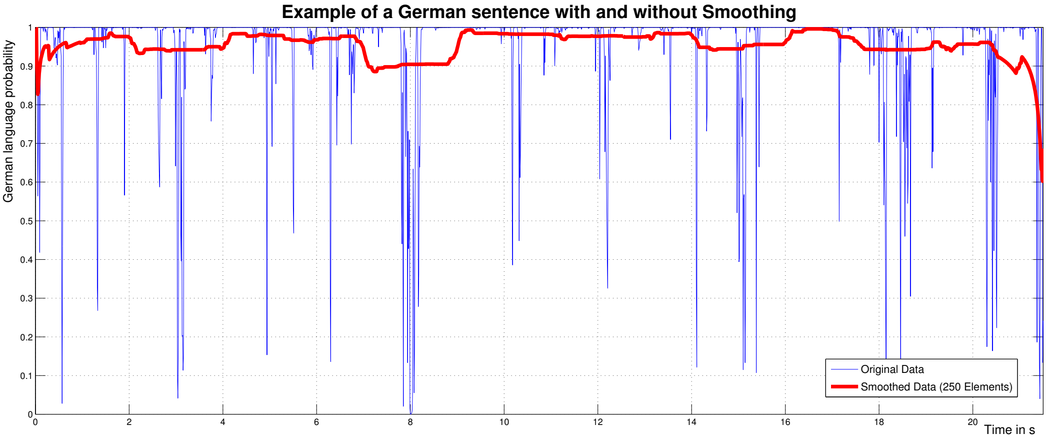
	Accuracy	
Approach	**Dev**	**Test**
Shared LID	99.5 %	98.7 %
Separate LID	99.0 %	99.3 %
Shared VLID	98.8 %	98.7 %

To get a better understanding on why some sentences are assigned the wrong language, we take a look at the misclassifications of the shared LID system. Some remarkable aspects are listed below:

- 78.6 % of the sentences are wrongly recognized as French. This might be explained by German speakers speaking more often in French than the inverse case.
  - 30.3 % of those are of the form "Danke, Herr ... ." ("Thank you, mister" + job title). These sentences are extremely short and therefore hard to classify. They are also used by French speakers, which might make it harder for our system to be trained appropriately.
  - 12.1 % of those include mumbling that is missing in the transcription. This never appears in the French sentences and might be due to the transcription having been done by mostly French natives.
- 21.4 % of the sentences are wrongly recognized as German.
  - 33.3 % of those are short sentences with less than 5 words.
  - 33.3 % of those are missing one syllable. All those cases are spoken by foreign speakers.

The multiplication metric has the disadvantage that statistical outliers can lead to very bad results. To get a smoother and less error prone result, we devise other metrics.

Moving Average We try different values of the windows size N to achieve the best possible results. The output is smoother than the input and our classifier becomes more stable. Unfortunately the first and last d elements are not equally smooth as can be seen in Figure 5.1.3, because there are less neighboring elements.



Hidden Markov Model To optimize the performance of our HMM we can vary a number of parameters. These are the number of states, that describes how many states the HMM has, the moving average windows size that defines on how many elements we apply a moving average, the simplification threshold that drops language sequences that occur for a duration of less than a predefined threshold, the language model scale, that defines the importance of the specified language model and the word insertion penalty which can be varied to increase or decrease the number of insertions.

We try several combinations of these parameters to find a local optimum. An appropriate choice of states in the HMM dominates over the window size and simplification threshold, hence we do not consider the other parameters. The remaining two free parameters can now be plotted to find a local optimum. The following figure shows the VLID accuracy (color and size of the dots) for a given word insertion penalty (x-axis) and LM scaling factor (y-axis). We can see that a higher LM scaling factor typically requires a higher word insertion penalty. A local maximum can be found at $(100, -80)$. We follow the same procedure for the shared LID system and freeze the parameters for testing.

We use the tuned parameters from 5.1.3 to evaluate our test set in sentences and interventions. The following table shows the results. Sentence-

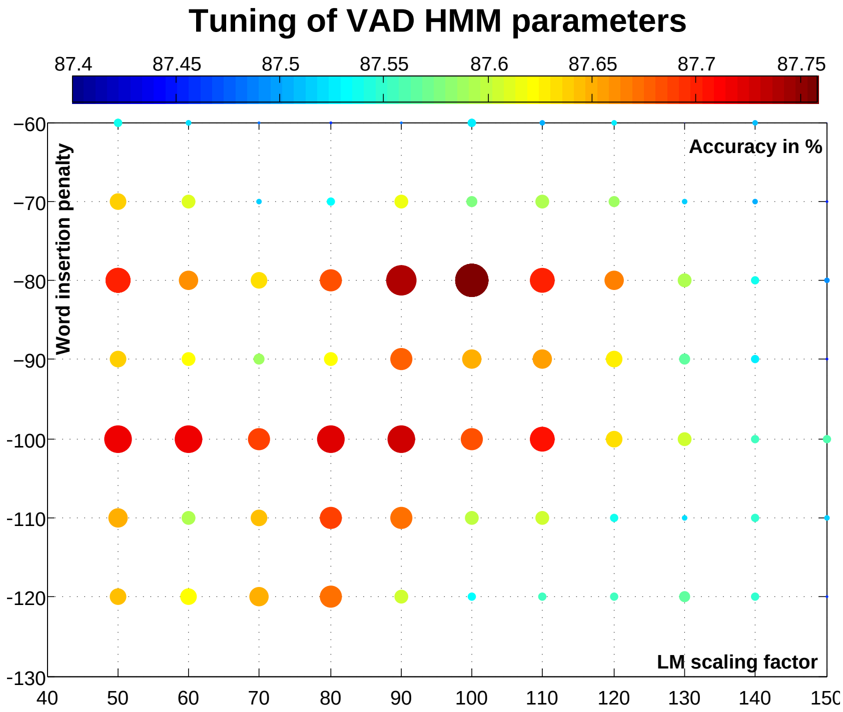


Figure 12: Tuning of parameters

based accuracy refers to the percentage of sentences that are correct whereas time-based accuracy takes into account the duration of the sentences. The performance of the HMM is better than in the case of multiplication. The significant performance difference between sentences and interventions can be explained by code-switches in the interventions and inaccuracies in the segmentation. The shared recognizer LID approach performs significantly better than any other approach. The separate recognizer LID has the worst performance.

Table 9: Language Identification results with HMM metric

		Accuracy	
Approach	**Dataset**	**Time-based**	**Sentence-based**
Shared LID	Sentences	99.5 %	98.7 %
Shared VLID	Sentences	99.6 %	98.8 %
Shared Recognizer LID	Sentences	100.0 %	100.0 %
Shared Recognizer4 LID	Sentences	99.8 %	98.1 %
Separate Recognizer LID	Sentences	95.2 %	94.2 %
Shared LID	Interventions	83.3 %	-
Shared VLID	Interventions	87.8 %	-

5.2 Automatic Speech Recognition

5.2.1 Monolingual ASR

Tables 25 and 26 show the word accuracies of the ASR decoding on German and French respectively. It can be seen that the German decoding is about 10 % more accurate than the French one.

To understand this we look at the output of the decoders and compare it to the original transcription:

Sentence 1: The following is a German sentence with 50 % word accuracy:

LAB[8]: gilbert loretan wurde neunzehn hundert drei und sechzig geboren verheiratet ist er mit brigitte albrecht ehemalige spitzenlangläuferin und er i und vater von céline

REC: sie herr loretan wurde neunzehnhundert dreiundsechzig geboren verheiratet ist er mitglied albrecht ehemalige spitzenlangläuferin in der und vater von fällen

Let us take a look at misrecognized words:

- All 3 French first names are not recognized. This can be explained by the different pronunciation in both languages.
- The number "1963" is expressed as a concatenation of numbers (1900 + 63), but acoustically it is indifferentiable from another sequence of numbers (19 + 100 + 3 + 60). Therefore the segmentation of words is wrong and deteriorates our word accuracy.
- The non-word "i" can be explained by mumbling in the recording which was not correctly transcribed as such.

Sentence 2: Another German sentence with 89 % word accuracy:

LAB: die c. s. p. o. fraktion ist der meinung dass nach wie vor aus regionalpolitischen gründen gewisse entscheidungen vielleicht sinnvoll sein mögen was aber gesundheitspolitisch nicht konsequent ist

REC: die zuerst p. o. fraktion ist der meinung dass nach wie fahrer aus regionalpolitischen gründen gewisse entscheidungen heute vielleicht sinnvoll sein mögen was aber gesundheitspolitisch nicht konsequent ist

- This sentence was expressed very clearly by a native German speaker and we can see that most of the words are recognized well.
- However the recognition of abbreviations is a very hard task. The letters c and s (/t s e: Q E s/) in the abbreviation are recognized as the word "zuerst" (/t s u Q e: 6 s t/).

[8] LAB denotes the annotated ground truth and REC refers to the decoder output

- The appearance of the word "heute" in the output of the decoder does not seem plausible from an acoustic point of view, since there is no pause or mumbling in that place, which might be misinterpreted as that word. It could instead be due to the LM assigning higher probabilities to the sequence "entscheidungen heute" (P_1) and "heute vielleicht" (P_2) than to "entscheidungen vielleicht" (P_3). Indeed, a look at the log probability reveals that $P_1 = -3.9$ and $P_2 = -7.5$, whereas $P_3 = -8.9$. So in this case the evidence that was collected from the data, that the case of P_1 is very common is actually misleading and leads to a mistranscription.

Sentence 3: French sentence with 75 % word accuracy:
LAB: le projet de décision prend en compte l' hypothèse la plus défavorable pour le canton concernant ce taux de subventionnement
REC: le projet de décision rencontre hypothèses de la plus défavorable pour le canton concernant ce taux de subventionnement

- Due to acoustic similarity, the words "prend en compte" are recognized as "rencontre".
- Many plural words in French are pronounced in the same way as the singular form and are therefore impossible to differentiate.
- Words that exist of one or two phonemes ("l'", "de") are often attached to neighboring words or simply skipped since there is not much probabilistic evidence for such a short word.

Table 10 shows the performance of the monolingual systems. We can see that German development data performs about 10 % better than any other data. There can be various reasons for this observation. The German system has about 50 % more phonemes than the French system, which could lead to more discriminative MLP outputs. The fact that all bilingual speakers are in the test set might explain the limited generalizability of the German recognizer.

Table 10: Monolingual ASR performance

Language	**Dataset**	**Accuracy**
German	Sentences (DEV)	77.8 %
German	Sentences (TST)	67.0 %
French	Sentences (DEV)	68.1 %
French	Sentences (TST)	66.4 %

5.2.2 Bilingual ASR

In this section we compare the performance of Systems 1 to 4. The results show that System 2 performs significantly better than any other system. In fact, the performance is even close to the oracle LID solution. This can be explained by the high LID performance on segmented sentences. This shows that LID can improve the performance of multilingual ASR systems. Furthermore, the linear combination of phone class posteriors as used in System 4 has a higher performance than System 1. However we also see that monolingual speech recognizers outperform joint multilingual recognizers. The higher perplexity for the multilingual language model used in Systems 1 and 4 and the higher number of words in the dictionary also lead to a decrease in performance.

The results for the integration of LID into ASR are comparable to what can be found in the literature. [WeinerVu12] report a 4 % relative improvement in Mixed Error Rate (MER) on bilingual ASR (English and Mandarin) when LID has a minimum frame accuracy of 85 %. A multistream approach is used to combine acoustic model score and language information at frame level. This is combined with a technique called "language lookahead“ and applied onto a corpus with intra-sentential code-switches.

Table 11: Bilingual ASR performance

	Accuracy	
System	**Dev**	**Tst**
System 1	65.8 %	59.6 %
System 2	71.9 %	66.5 %
System 3	69.3 %	63.6 %
System 4	68.5 %	61.7 %
Oracle LID	72.9 %	66.8 %

5.3 Discussion

In this section we investigate higher level language properties such as accented speech and nativeness, code-switches, sentence duration and channel properties.

5.3.1 Accented Speech

Figures 5.3.1 and 5.3.1 show two monolingual German utterances. One is spoken by a native German and the other by a native French speaker. Both utterances were evaluated using the same setup (shared phone LID, averaging over 250 elements), but the contents of the sentences are different. Nevertheless these utterances show a trend which seems to be prevailing throughout most of the data. The language identification constantly works well on native speakers, whereas there are considerable outbreaks on non-native speakers. We can even see from the data if a speaker can be considered as fluent speaker and without a strong accent in both languages.

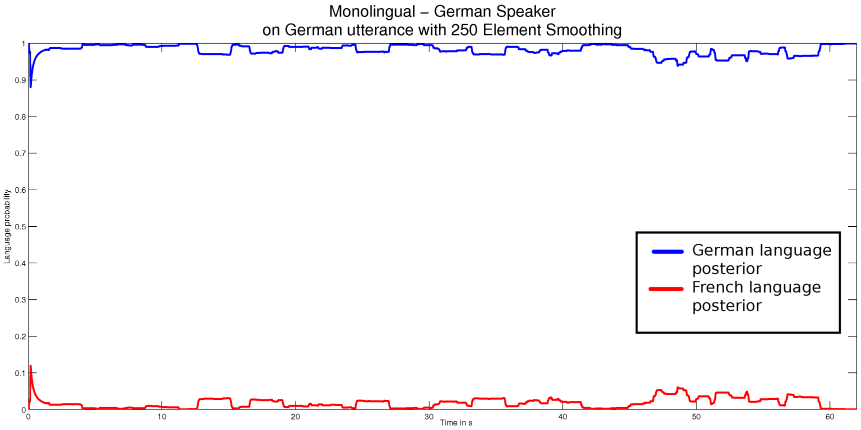


Figure 13: Example of a native German utterance

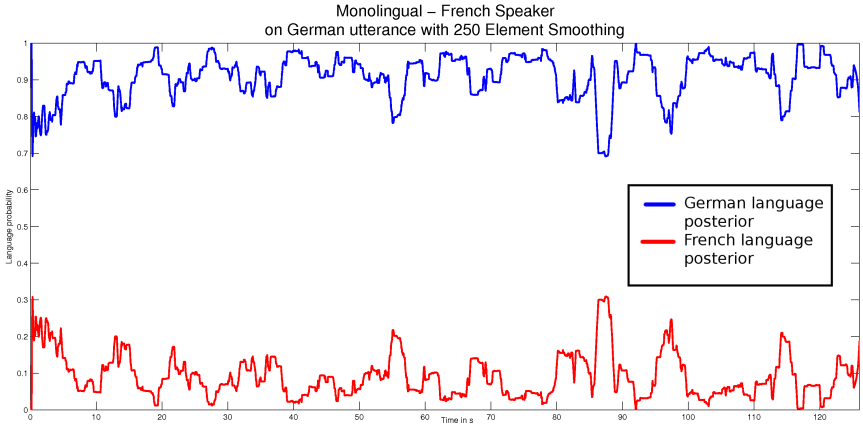


Figure 14: Example of a non-native German utterance

5.3.2 Native versus Non-native

We rescore our LID and ASR systems on native and non-native speaker data to see if there is a significant difference between both. Table 12 shows that the results on the shared LID system are very similar between native and non-native speakers, but that on shared VLID we get a 1.6 % drop in accuracy. In case of the separate recognizer LID the performance drops by about 15 %. The decrease in performance on the native data can be explained by the perplexity mismatch of the different languages, that may bias the recognizer towards one language (see Section 3.3), whereas the low performance on non-native data seems to be caused by the acoustic mismatch on non-native utterances.

The ASR results in Table 13 show a 11.1 % drop from native to non-native on French data and a 1.4 % drop for German data.

Table 12: Effect of nativeness on bilingual LID accuracy

	LID Accuracy		
Approach	**Native**	**Non-Native**	**Overall**
Shared LID	98.8 %	98.3 %	98.7 %
Shared VLID	99.1 %	97.5 %	98.8 %
Shared Recognizer LID	100.0 %	100.0 %	100.0 %
Separate Recognizer LID	96.3 %	81.5 %	94.2 %

Table 13: Effect of nativeness on monolingual ASR accuracy

	ASR Accuracy		
Language	**Native**	**Non-Native**	**Overall**
French	70.1 %	59.0 %	68.1 %
German	66.9 %	68.3 %	67.0 %

Hence we can conclude that non-native speech is considerably more difficult for both LID and ASR.

5.3.3 Speaker Dependency

We have shown that the LID and ASR performance is dependent on the nativeness. Now we analyze the performance on a per speaker level. By listening to the recordings we categorize the 7 speakers in the test set as native German, bilingual and native French. Table 14 shows the LID and VLID performance for each speaker. The last two columns show the difference of French and German LID performance. The sign is important. Positive values indicate that French LID (blue) performs better and negative (green) values indicate that German performs better.

The results indicate that, except for speaker 094, the nativeness of a monolingual speaker can be measured. It also shows which speaker can be considered as bilingual from an acoustic perspective.

Table 14: Comparison of speaker dependent LID accuracy

		French Data		**German Data**		**Difference**	
Spkr	**Lang.**	**LID**	**VLID**	**LID**	**VLID**	**ΔLID**	**ΔVLID**
059	German	93.6 %	90.3 %	100.0 %	100.0 %	-6.5 %	-9.7 %
079		95.5 %	90.9 %	98.1 %	99.6 %	-2.7 %	-8.7 %
191	Biling.	99.4 %	96.4 %	98.7 %	100.0 %	0.7 %	-3.6 %
109		100.0 %	98.7 %	98.8 %	99.8 %	1.2 %	-1.0 %
094	French	99.7 %	97.1 %	97.2 %	100.0 %	2.5 %	-2.9 %
096		100.0 %	100.0 %	87.5 %	87.5 %	12.5 %	12.5 %
102		98.6 %	98.6 %	71.4 %	85.7 %	27.2 %	12.9 %
Avg.		99.4 %	97.4 %	98.4 %	99.7 %	1.0 %	-2.2 %

Table 15 shows the number of sentences and their cumulative duration with respect to speaker and language. It also presents the duration after subtracting silence. We can see that the amount of silence is balanced with respect to the speaker. Thus Table 15 confirms that the results are not unreliable due to excessive silence.

Table 16 shows the speaker dependent ASR performance on German and French sentences. On French sentences, there is a 28 % difference in ASR performance between maximum and minimum. We can see in Table 15 that few non-native German sentences exist for speakers 096 and 102. This could make the respective numbers in 16 less meaningful.

Table 15: Comparison of speaker dependent sentence duration

	French			German		
Speaker	**#Sent**	**duration**	**dur$_{VAD}$**	**#Sent**	**duration**	**dur$_{VAD}$**
059	31	281s	258s	195	2133s	1999s
079	22	148s	137s	698	4873s	4462s
191	166	1993s	1876s	310	3928s	3748s
109	233	3121s	2968s	402	4669s	4475s
094	313	3910s	3686s	72	770s	670s
096	91	1249s	1191s	8	77s	73s
102	72	719s	672s	7	29s	25s

Table 16: Comparison of speaker dependent ASR accuracy

	ASR Accuracy	
Speaker	**German**	**French**
059	75.0 %	51.5 %
079	65.4 %	46.7 %
109	70.8 %	64.5 %
191	59.8 %	53.3 %
094	68.5 %	71.8 %
096	78.0 %	75.2 %
102	61.4 %	73.9 %
Avg	67.0 %	66.4 %

5.3.4 Sentence Duration

We already discussed before that the performance of our LID or VLID systems depends mostly on the duration of the sentences and that shorter sentences are harder to detect. To support that claim, we measure the duration of a sentence compared to the VLID accuracy. Since the accuracy of one sentence is binary (right or wrong), we compare the duration of all correct and all incorrect sentences. Table 17 gives us an overview of the sentence duration of the shared VLID test sentences. We can see that on average, sentences that are not assigned the correct language, are almost a third of the duration of those that are correct. We can also see that no sentence that is longer than 8.5s has been misrecognized, even though there are sentences as long as 70.5s.

Table 17: Analysis of time vs. accuracy of shared VLID

	Duration		
Sentences	**Average**	**Minimum**	**Maximum**
Correct VLID	10.7s	0.6s	70.5s
Incorrect VLID	3.7s	0.8s	8.5s

Figure 15shows the relation of shared VLID accuracy and sentence duration. To be able to visualize this, we average the sentence duration over 65 sentences (sorted by duration) and plot the average LID accuracy. We can see that any sentence that is longer than 9s is recognized correctly and that the accuracy increases with sentence duration.

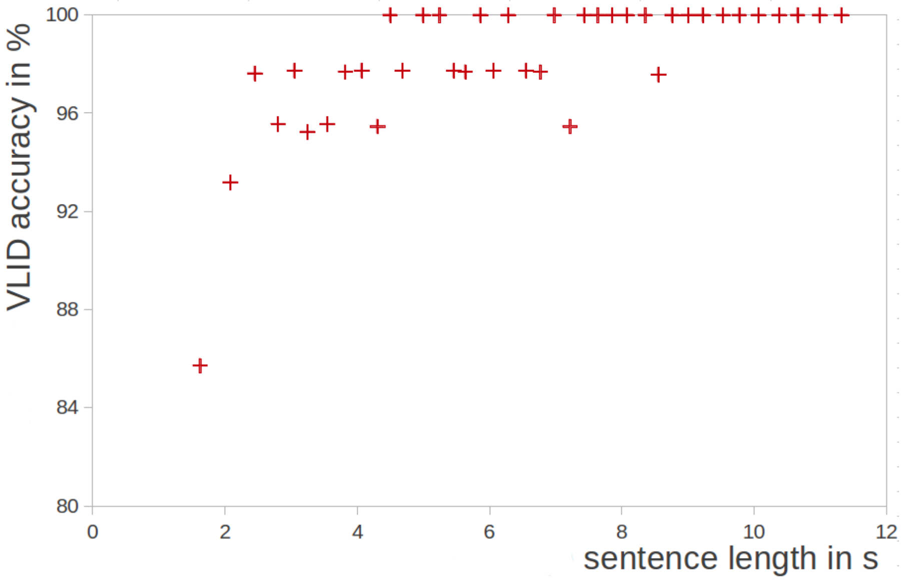


Figure 15: VLID accuracy with respect to sentence length

5.3.5 Channel Properties

As mentioned in chapter 3.1, the audio data for the MediaParl speech corpus has been recorded in the years 2006 and 2009. Therefore we have to take into account that different channel properties can have a different effect on the performance of our systems. Table 18 shows the performance of monolingual ASR and shared phone VLID with respect to the year of recording. We can see significant variations in monolingual ASR accuracy. For both languages, we have a higher accuracy on the dev set for 2006 and on the test set for 2009. Differences in sampling rate, audio encoding (see Section 3.1) and the amount of available data per year can be reasons for this effect.

Table 18: Comparison of accuracy of 2006 and 2009

	Accuracy	
Dataset	**Dev**	**Test**
2006 German ASR	78.8 %	64.5 %
2009 German ASR	76.0 %	69.4 %
2006 French ASR	68.6 %	67.9 %
2009 French ASR	67.9 %	64.9 %
2006 VLID	99.5 %	99.3 %
2009 VLID	98.5 %	98.4 %

5.3.6 Code Switches

Given the information from the previous sections we can now analyze the code-switches in our speech corpus. Figure 5.3.6 shows a bilingual utterance where a German speaker switches from German to French and back. The black vertical bars represent the ground truth of the code-switch. In theory, any time at which the red and blue graph cross each other should represent a code-switch. It can be seen that there are long sections (i.e. about 19 seconds) where the language is correctly detected. But there are also problems with our approach of detecting code-switches:

- As could be seen in the previous section, accented speech deteriorates the detection performance. Therefore the French part has too low probabilities and it looks as if there where many more code-switches during that section

- The timing is incorrect, especially with the second code-switch that is about 4 seconds away from where it should be. This might be a negative side effect of the smoothing which is done on the data, but that cannot be the only reason as the effect is not visible on the first code-switch, even though the smoothing filter is symmetric.

- Figure 5.3.6 shows a similar situation for a French speaker. There are more wrongly detected switching points. They could be eliminated by setting a higher smoothing threshold, but that would also decrease the number of detected code-switches. Humans can clearly see that the characteristics on the French part are completely different from those on the German part.

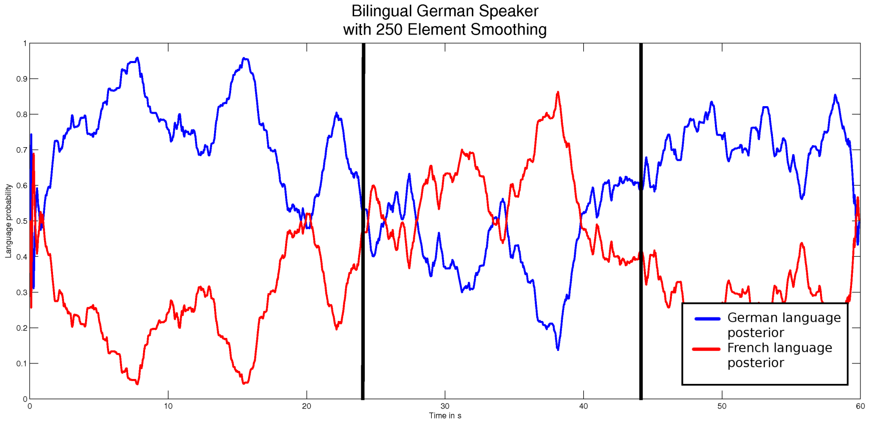


Figure 16: Example of a bilingual utterance spoken by a German native

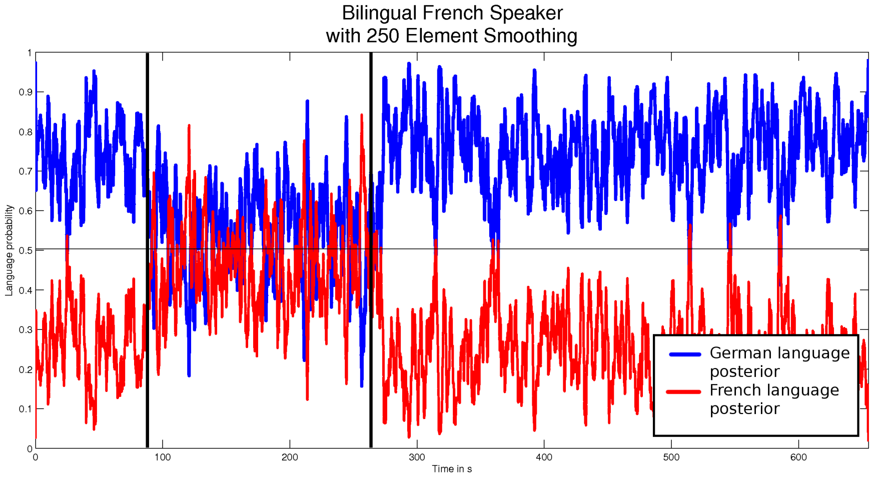


Figure 17: Example of a bilingual utterance spoken by a French native

5.3.7 Future work

Future work at IDIAP will further investigate how to process unsegmented interventions. Preliminary studies have performed very badly on this task. We applied Voice Activity Detection with the ASR systems described above and enforced a minimal speech duration of 5s. We then split the interventions at the silence parts and passed the resulting segments forward through the phone MLPs and language MLPs.

As already seen, the LID performance is about 15 % absolute worse. Lower LID performance especially effects Systems 2 and 3. Furthermore, a minimal duration of 5s is less than half of the length of the annotated sentences. Hence a sentence will typically contain several segments. During decoding however, we assume that each segment is one sentence. This might obviously cause degradation.

6 Conclusion

The purpose of this bachelor thesis was to investigate how LID can improve ASR. We studied several systems and their performance on the MediaParl speech corpus from the IDIAP research institute. Chapter 2 presented the theory of ASR and LID using ANNs. In Chapter 3 we elaborated on the problems of creating a pronunciation lexicon and pointed out that each language has its own characteristics that have to be taken into account. In Chapter 4 we presented three hierarchical MLP based approaches and two ASR-dependent LID approaches. The approaches were evaluated and we found that Voice Activity Detection can be helpful for more accurate results and to be able to do segmentation. Nevertheless, the recognizer LID approaches outperformed the hierarchical approaches due to higher order information such as the language model.

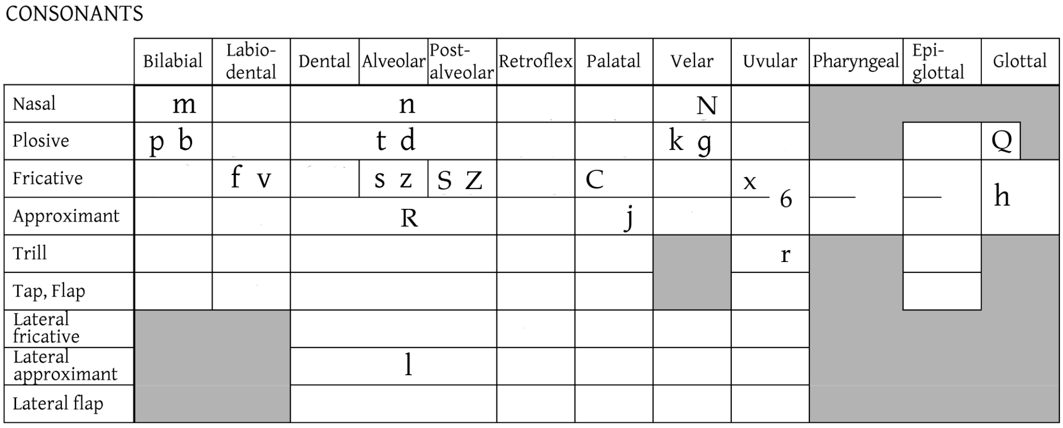
We presented four bilingual ASR systems and found out that a combination of reliable Language Identification and monolingual ASR systems provides the best performance. Finally, we analyzed the influence of nativeness, speaker dependency, sentence duration, channel properties and code-switches on our systems.

It could be shown that LID can improve ASR and that ASR can be used for LID. Future work will consist of processing longer unsegmented utterances, while focusing on code-switches.

7 Appendix

7.1 Phoneme Sets

German Phonemes



CONSONANTS

	Bilabial	Labio-dental	Dental	Alveolar	Post-alveolar	Retroflex	Palatal	Velar	Uvular	Pharyngeal	Epi-glottal	Glottal
Nasal	m		n					N				
Plosive	p b		t d					k g				Q
Fricative		f v		s z	S Z		C		x			h
Approximant			R				j		6			
Trill									r			
Tap, Flap												
Lateral fricative												
Lateral approximant			l									
Lateral flap												

Figure 18: German consonants

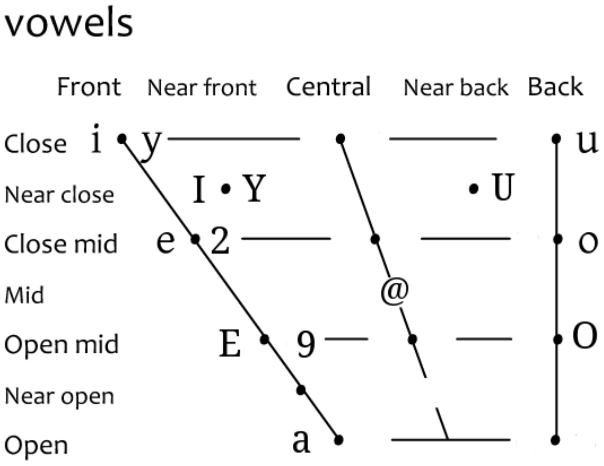
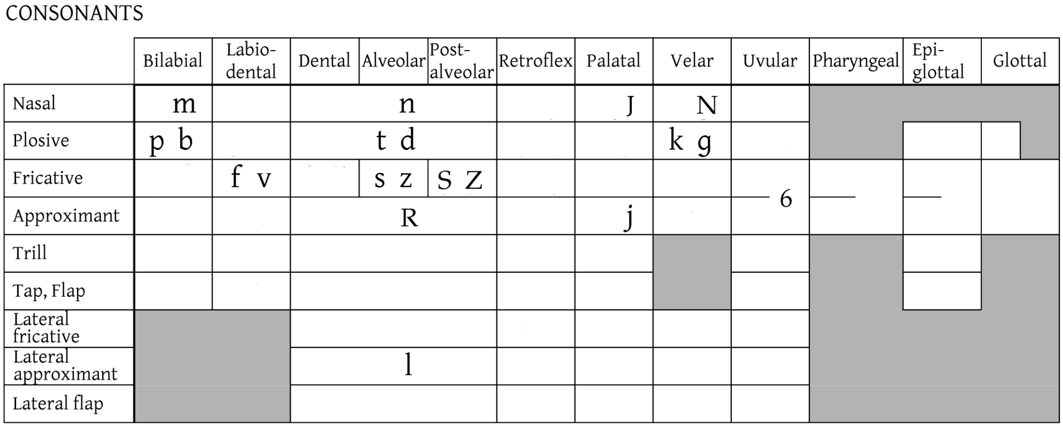


Figure 19: German vowels

Other Phonemes Silence: sil
Stretched: i:, y:, u:, e:, 2:, o:, E:, O:, a:
Nasals: e∼, o∼, E∼, a∼
Nasal and streched: o∼:, E∼:, a∼:
Diphthongs: OY, aU, aI

French Phonemes



CONSONANTS

	Bilabial	Labio-dental	Dental	Alveolar	Post-alveolar	Retroflex	Palatal	Velar	Uvular	Pharyngeal	Epi-glottal	Glottal
Nasal	m			n			J	N				
Plosive	p b			t d				k g				
Fricative		f v		s z	S Z				6			
Approximant				R			j					
Trill												
Tap, Flap												
Lateral fricative												
Lateral approximant				l								
Lateral flap												

Figure 20: French consonants

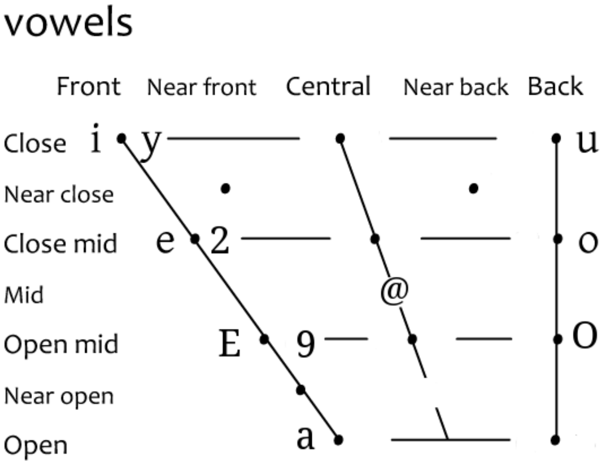


Figure 21: French vowels

Other phonemes Silence: sil
Labial-pallatal approximant: H
Voiced labio-velar approximant: w
Nasals: e∼, o∼, a∼, 9∼

7.2 Training Results

Table 19: Separate phone MLP training results for German

Epoch	Train accuracy	CV accuracy	Learning rate
0	-	01.0 %	-
1	58.1 %	62.9 %	0.0008
2	68.4 %	68.0 %	0.0008
3	70.7 %	69.1 %	0.0008
4	71.9 %	70.0 %	0.0008
5	72.7 %	68.9 %	0.0008
6	74.1 %	70.9 %	0.0004
7	74.7 %	71.5 %	0.0002
8	75.0 %	71.8 %	0.0001

Table 20: Separate phone MLP training results for French

Epoch	Train accuracy	CV accuracy	Learning rate
0	-	03.8 %	-
1	61.4 %	62.3 %	0.0008
2	71.2 %	63.8 %	0.0008
3	73.0 %	66.1 %	0.0008
4	74.0 %	65.9 %	0.0008
5	75.4 %	68.6 %	0.0004
6	76.0 %	69.5 %	0.0002
7	76.3 %	69.7 %	0.0001

Table 21: Shared phone MLP training results for German and French

Epoch	Train accuracy	CV accuracy	Learning rate
0	-	02.3 %	-
1	56.3 %	52.7 %	0.0008
2	65.4 %	58.6 %	0.0008
3	67.4 %	59.8 %	0.0008
4	68.3 %	60.6 %	0.0008
5	69.3 %	61.5 %	0.0008
6	69.9 %	63.5 %	0.0008
7	70.4 %	63.0 %	0.0008
8	71.7 %	64.4 %	0.0004
9	72.2 %	65.7 %	0.0002
10	72.5 %	66.1 %	0.0001

Table 22: Language MLP Training results for shared phone LID

Epoch	Train accuracy	CV accuracy	Learning rate
0	-	50.2 %	-
1	94.6 %	95.9 %	0.0008
2	98.3 %	96.3 %	0.0008
3	98.8 %	97.0 %	0.0004
4	99.1 %	97.2 %	0.0002

Table 23: Language MLP Training results for separate phone LID

Epoch	Train accuracy	CV accuracy	Learning rate
0	-	49.4 %	-
1	93.4 %	95.3 %	0.0008
2	98.3 %	96.2 %	0.0008
3	98.9 %	96.5 %	0.0008
4	99.2 %	97.0 %	0.0004

Table 24: Language MLP Training results for VLID

Epoch	Train accuracy	CV accuracy	Learning rate
0	-	31.7 %	-
1	94.8 %	96.0 %	0.0008
2	98.2 %	96.7 %	0.0008
3	98.6 %	96.9 %	0.0004
4	99.0 %	97.2 %	0.0002

Table 25: ASR performance on German

LM scale	Word penalty	Accuracy
3	2	73.2 %
5	-4	77.3 %
5	-3	77.4 %
5	-2	77.5 %
5	-1	77.6 %
5	0	77.8 %
5	1	77.7 %
5	2	77.7 %
5	5	77.0 %
10	-10	69.5 %
10	10	74.3 %

Table 26: ASR performance on French

LM scale	Word penalty	Accuracy
2	2	51.4 %
4	4	64.7 %
5	-4	67.3 %
5	-2	67.8 %
5	-1	68.1 %
5	0	68.1 %
5	1	67.8 %
5	2	67.5 %
6	-2	67.3 %
6	6	65.7 %
8	8	64.3 %

Table 27: ASR performance on joint German-French

LM scale	Word penalty	Accuracy
4	4	60.3 %
5	-3	65.7 %
5	-2	65.8 %
5	-1	65.7 %
5	0	65.5 %
6	-2	65.3 %
10	10	57.2 %

References

[Bengio96] Y. Bengio, *Neural Networks for Speech and Sequence Recognition.* International Thomson Computer Press, 1996.

[BisaNey04] M. Bisani and H. Ney, *Bootstrap estimates for confidence intervals in ASR performance evaluation.* in Proc. of ICASSP, pp. I–409–412, 2004.

[Cybenko89] G. Cybenko, *Approximations by superpositions of sigmoidal functions*, Mathematics of Control, Signals, and Systems, 1989.

[FuegenSchultz03] C. Fügen, T. Schultz, S. Stüker, H. Soltau, F. Metze *Efficient Handling of Multilingual Language Models*, Proceedings of the Workshop of Automatic Speech Recognition Understanding, St. Thomas, 2003.

[Hermansky89] H. Hermansky, *Perceptual Linear Predictive (PLP) analysis of speech*, Speech Technology Laboratory, Division of Panasonic Technologies, Inc., 1989.

[HTK06] S. Young, G. Evermann, *The HTK Book* Cambridge University, Department of Engineering, 2006.

[Imseng10] D. Imseng, M. Magimai.-Doss, H. Bourlard, *Hierarchical Multilayer Perceptron based Language Identification*, Proceedings of Interspeech, 2010.

[IPA99] International Phonetic Association, *Handbook of the International Phonetic Association: a guide to the use of the international phonetic alphabet*, Cambridge University Press, 1999.

[Kumar10] P. Kumar, H. Li, R. Tong, P. Matejka, L. Burget, J. Cernocky *Tuning Phone Decoders for Language Identification* International Conference on Acoustics, Speech and Signal Processing, 2010.

[Nilep06] C. Nilep, *Code Switching in Sociocultural Linguistics*, University of Colorado, Bolder, 2006.

[Robinson96] A.J. Robinson, *Speech Analysis*, 1996.

[Rosenfeld97] R. Rosenfeld, *Statistical Language Modeling and N-grams*, 1997.

[Schultz95] T. Schultz, I. Rogina, A. Waibel, *Experiments with LVCSR Based Language Identification* Proceedings of the Speech Research Symposium SRS XV, pp 89-94, 1995.

[Schultz06] T. Schultz, K. Kirchhoff, *Multilingual Speech Processing*, Academic Press, 2006.

[SIL92] R. G. Gordon, *Ethnologue: Languages of the World*, SIL International, 12th Edition, 1992.

[VuKrausIS11] N. Thang Vu, F. Kraus, T. Schultz, *Rapid building of an ASR system for Under-Resourced Languages based on Multilingual Unsupervised Training*, Interspeech, 2011.

[VuKraus11] N. Thang Vu, F. Kraus, T. Schultz, *Cross-Language Bootstrapping Based on Completely Unsupervised Training Using Multilingual A-Stabil*, International Conference on Acoustics, Speech and Signal Processing, 2011.

[WangSchultz03] Z. Wang, T. Schultz, *Non-Native Spontaneous Speech Recognition through Polyphone Decision Tree Specialization*, Eurospeech, 2003.

[WeinerVu12] J. Weiner, N. Thang Vu, D. Telaar, F. Methe, T. Schultz, D. Lyu, E. Chng, H. Li, *Integration of Language Identification into a Recognition System for Spoken Conversations Containing Code-Switches*, SLTU, 2012.

[Young96] S. Young, *Large Vocabulary Continuous Speech Recognition: a Review*, Cambridge University Engineering Department, 1996.

[Zissman01] Marc A. Zissman, Kay M. Berkling, *Automatic Language Identification*, Information Systems Technology Group, Lincoln Laboratory, Massachusetts Institute of Technology, 2001.