Karlsruhe Institute of Technology

Automatic Language Identification for Natural Speech Processing Systems

Student Research Paper of

Michael Heck

At the Department of Informatics
Institute of Anthropomatics (IFA)
Interactive Systems Laboratories (ISL)

Supervisors:
Prof. Dr. Alex Waibel
Dr. Sebastian Stüker

Duration: 01. June 2011 – 01. September 2011

Abstract

In this work we describe the realization of selected standard approaches to automatic language identification and subsequent improvements. The two approaches parallel phone recognition (PPR) and parallel phone recognition followed by language modeling (PPRLM) will be investigated in detail by means of theoretical observation and implementation.
We first build PPR and PPRLM baseline systems that produce score-fusing language cue feature vectors for language discrimination and utilize an SVM back-end classifier for the actual language identification. Experiments on our German and English bi-lingual lecture tasks reveal, that the PPRLM system clearly outperforms the PPR system in various segment length conditions, however at the cost of slower run-time.
We show ways of improving the baseline systems by using lexical information in the form of keyword spotting in the PPR framework, and by the incorporation of additional language models in the PPRLM architecture. In order to combine the faster run-time of the PPR system with the better performance of the PPRLM system we finally build a hybrid of both approaches that clearly outperforms the PPR system while not adding any additional computing time. Finally we make suggestions for future work including the use of our results in the KIT lecture translation system.

Acknowledgements

First of all I would like to thank my supervisor Sebastian Stüker for his constant support and guidance during this project. With his advice and explanations he helped me to learn a lot about the field of Natural Language Processing and scientific methods in general. I would also like to thank Alex Waibel for giving me the opportunity to do the research for this report at the Interactive Systems Lab. Thanks go to Florian Kraft and Christian Saam for introducing me to the database. Further my thanks go to Christian Mandery for his help with SRILM and the Janus Recognition Toolkit. Finally I would also like to thank Christian Fügen, Tina Milo, Rainer Saam and Kevin Kilgour for all the interesting discussions.

Contents

1. Introduction

In recent years automatic language processing technologies have seen large improvements in terms of performance, use and acceptance. Speech recognition and speech-to-speech translation systems manifest themselves in a large variety of applications. In a globalizing world and growing multi-cultural societies one of the most important requirements to spoken language technology is the ability to cope with multiple languages in a robust and natural fashion. In order to achieve these objectives, technologies are required that enable the easy use of language without being impeded by language barriers. Today's smart systems are capable of multi-lingual speech processing, but usually it is the user's turn to decide which language pairs or pools should be worked on. Even the most sophisticated speech-to-speech and speech-to-text translation systems generally precede, besides the step of choosing one or more target languages, the explicit declaration of the source language. In our opinion, a highly desirable feature for intuitive and self-explanatory speech processing applications is the automation of this additional step, as it is a natural process in human-to-human interaction to identify the language, that was chosen to enter into a dialogue with, without asking. In the past decades a vivid interest grew in automatizing language identification with help of well-established speech processing technologies. The associated field of research is referred to as automatic language recognition or, more commonly, as *automatic language identification (LID)*. The main idea is to especially utilize and exploit the sources of information, which are essential in the human language identification process.

1.1 Language Identification

The task of automatic language identification is the machine-made identification of the language used by an unknown speaker for voicing an utterance [AD10]. Humans are capable of recognizing almost immediately the language being spoken, and even if the used language is unknown, a human listener is often capable of categorizing it into a language family or similar classes. Humans make use of different sources of information, where the most important are: Acoustics, phonetics, phonotactics and prosody as well as lexical and morphological knowledge. The major questions of LID include: How do humans proceed in identifying a language? Which levels of information are most promising for effective language identification? Which approaches are best for a given type of application? Which language specific resources are required for spoken language modeling? How strong is the correlation between signal duration and identification accuracy [AD10]? For LID by machines the signal processing is very similar to those of other language processing tasks

like speech recognition and make use of the same acoustic features. A voiced utterance is digitalized with use of appropriate recording equipment, followed by pre-processing steps for transforming the signal into a sequence of acoustic feature vectors. Although certain systems already perform a classification on those feature vectors, usually a speech decoder tries to find the most probable sequence of sound units, i.e. phones, according to the observed sequence. A language decoder scores this sequence given a set of potential target languages. A back-end classifier performs an optimization of the final output and does the classification given the obtained language cues, e.g., acoustic or language model scores, and a classification criterion.
The main functional requirement for an LID system is a classification performance, i.e. identification accuracy, at the best possible rate, given an input signal of minimal length. Achieving the least computational complexity possible is of importance for embedded systems and real-time demanding applications. The acoustic-phonemic and phonotactic systems became the most popular approaches, being able to do robust identification on sufficiently short input sequences even for real-time demanding scenarios [AD10, Nav06]. Both approaches benefit from decades of research by computer speech scientists as well as linguists. Initial works on the field of automatic language identification, which were published in the seventies, observed the use of acoustic filter banks and documented first experiments with phonotactic constraints [AD10]. Later research focussed on the incorporation of linguistic concepts into LID by machines, such as the generation of phone models and, based on this, phonetic and phonotactic modeling, which are considered the most popular approaches to automatic LID up to the present day. Further research has been conducted on various other levels of information, such as prosody, morphology and syntax. The fusion of approaches is of growing interest in many of today's research projects. In the future, the issue of dialect and accent identification, as well as language boundary detection in cases of code-switching will further challenge the LID research society.

1.2 The JANUS Recognition Toolkit

The speech decoding modules of our systems are realized with the JANUS Recognition Toolkit (JRTk), which has been developed at the Karlsruhe Institute of Technology and Carnegie Mellon University as a part of the JANUS speech-to-speech translations system [FGH+97, LWL+97]. The toolkit provides an easy-to-use Tcl/Tk script based programming environment which gives researchers the possibility to implement state-of-the-art speech processing systems, especially allowing them to develop new methods and easily perform new experiments. JANUS follows an object oriented approach, forming a programmable shell. For this thesis, JRTk Version 5 was applied, which features the IBIS decoder. IBIS is a one-pass decoder, thus being advantageous with respect to real-time requirements of today's ASR and other language processing applications [SMFW01].

1.3 The SRI Language Modeling Toolkit

For the extraction of various language features the SRI Language Modeling Toolkit is used. SRILM is a toolkit for building and applying statistical language models, primarily for use in speech recognition, statistical tagging and segmentation, and machine translation. It has been under development in the SRI Speech Technology and Research Laboratory since 1995 [Sto02]. SRILM is a collection of C++ libraries, executable programs, and helper scripts designed to allow both production of and experimentation with statistical language models for speech recognition and other applications. The toolkit supports creation and evaluation of a variety of language model types based on n-gram statistics, as well as several related tasks, such as statistical tagging and manipulation of n-best lists and word lattices.

1.4 The LIBSVM Library

For performing the language classification on the basis of the obtained scores and cues we deploy the LIBSVM library. LIBSVM is an integrated software for support vector classification, regression and distribution estimation. It has been actively developed since the year 2000, with the goal to help users to easily apply support vector machines (SVMs) to their applications. LIBSVM has gained wide popularity in machine learning and many other areas, as it provides a simple interface where users can easily link it with their own programs. Main features of LIBSVM include among other things: Different SVM formulations, multi-class classification and various kernels [CL11].

1.5 Objective of This Work

This thesis addresses the theoretical concepts of automatic language identification and describes the realization of selected standard approaches and subsequent improvements. A special focus is on the processed average test segment length in order to determine which setup is most useful for a real-time demanding scenario, where fast and at the same time fairly accurate language identification is of high importance. The two approaches *parallel phone recognition (PPR)* and *parallel phone recognition followed by language modeling (PPRLM)* will be investigated in detail by means of theory and implementation of real systems. The aim is to lay the foundation for an LID system which is capable of being applicable as a front-end for several systems which would strongly benefit by the utilization of an LID pre-classifier. I.e., resulting systems should be able to do an accurate identification of the input language as additional information for a speech processing system, making manual language identification by a potential user obsolete. We hold the opinion, that the automation of this identification step greatly improves the usability and access to a huge variety of modern speech processing systems. We aim at a front-end application of our proposed LID systems for an existing automatic system for simultaneous speech-to-speech translation, which has been developed at the Karlsruhe Institute of Technology (KIT) [Füg09].
This thesis is organized as follows: Chapter 2 outlines basic principles of automatic language identification. An overview of the various levels of information, which take part in the characterization of languages will be given. We eludicate methods of performance measurement and describe a probabilistic formulation of a statistical approach to LID by machines. A summary of the well-established approaches is followed by an analysis of various design decisions. Chapter 3 provides a detailed insight into phonetic and phonotactic LID approaches, along with some linguistic background. The designs of our system implementations are explicated in Chapter 4. Following the introduction of the the dataset we are working on, the step-by-step realization our LID frameworks is described in detail. We proceed from a bilingual, closet set, forced decision case, namely German versus English. Chapter 5 delivers the experimental results of our test runs, in which we evaluate both baseline systems against each other. Further on, various system improvements are described, and we introduce an experimental PPRLM & PPR hybrid system, combining the previously developed systems to a new framework. The chapter is concluded by an Analysis of our results. Chapter 6 summarizes our work and gives an outlook on future work.

2. Automatic Language Identification

The LID task concerns the identification of the language used for voicing a message and is therefore different from similar tasks such as speaker identification, speaker verification or language detection, whose topic is the determination whether or not a speech event occurred. It is noteworthy that a significant difference between spoken LID and written LID exists, regarding the issue. At the same time all tasks mentioned above have certain concepts and approaches in common, whereby progress on these fields of research are advantageous for solving the language identification task.

2.1 Levels of Information

Each language has a characteristic structure which makes that language sound different from other languages. The language structure is defined by its phone inventory, syllable construction, phonotactic rules and prosodic patterns. *Phonotactics* describes the restrictions which combinations of phones are allowed to form syllables, morphemes or words, given a specific language. The differences in allowed combinations can be vast across languages. For example, Japanese has very strict phonotactic constraints which roughly restrict phone combinations to a consonant-vowel (C-V) pattern, in contrast to West Slavic languages, which even allow words being completely composed of consonants. *Prosody* is the rhythm, stress, and the intonation of speech. In terms of acoustics, the prosodics involve variation in syllable length, loudness, pitch, and the formant frequencies of spoken language.
Infants are not only able to recognize their future native language, but also to discriminate it from other, rhythmically different languages, as experiments showed [GM10]. Further research revealed that rhythmical differences were sufficient for discrimination and familiarity with the languages was not necessary. French newborns for example readily discriminated between utterances in English and Japanese, two languages they had never heard before [GM10].
Newborn infants also have surprising abilities to process acoustic information regarding word forms. They can detect the acoustic cues that signal word boundaries, i.e. typical stress patterns, the distribution of allophones or statistical regularities of phone co-occurrences, i.e. phonotactic constraints. They can discriminate words with different stress patterns and can distinguish between function words and content words on the basis of differing acoustic characteristics [GW08].
With aging it becomes more important whether a language is familiar to a listener or not,

due to the increased use of semantic information and metalinguistic knowledge [GW08, SMB96]. When listening to one's own language being spoken, the primary awareness is of the meaning of what was said [LM95]. Listeners attending to the semantic content are not aware of the phonetic form. But in the absence of meaning, only the acoustic-phonetic information is available. Several studies were carried out to evaluate the relative importance of various information levels in language discrimination by adults. Speech signals were more or less heavily transformed via speech re-synthesis or the shuffling of syllables to remove syntactic and semantic information from the signal and to gradually isolate broad phonotactics, rhythm and intonation. The results revealed, that the test subjects were able to distinguish languages on prosodic informations, given a certain amount of a priori knowledge about the target languages [RM99, AD10]. With broader knowledge of one or more target languages the already fair identification accuracy on phonetic and broad phonotactic information increases as well. Thus, we can assume that language identification by humans works with graceful degradation. The information of language identity is not exclusively encoded in one single level of information, but across different levels, as partly redundant information [Nav06].
The levels of information elaborated above are utilizable in automatic language identification. Figure 2.1 gives a short overview.

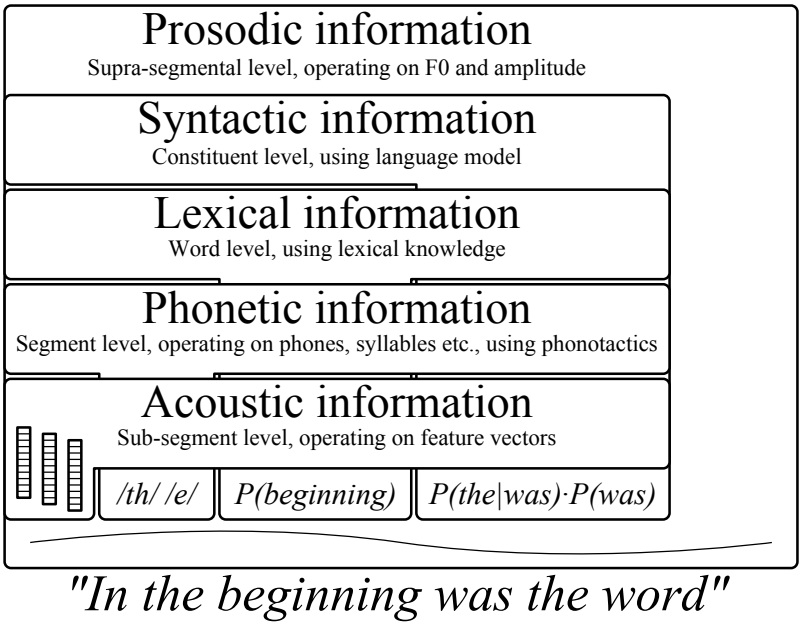

Figure 2.1: Overview of the utilizable levels of information for automatic language identification tasks.

Levels of information used in automatic language identification are:

Prosody
Prosodic information such as rhythm, stress and intonation manifest in the speech signal's fundamental frequency and the amplitude. Prosodics work on supra-segmental or sentence level.

Syntax
Syntactic information is approximable with language models, working on constituent level. Sentence patterns differ across languages, even when the pool of words is not disjunct.

Lexical information
Identification of individual words belonging to a specific language is working on word

level.

Phonetics & Phonotactics
Identification strategies working on phonemes, syllables and sub-words work on segmental level.

Acoustics
The lowest level is the signal level. Acoustic informations manifest in acoustic feature vectors generated upon a signal. Differences in acoustic features may roughly represent phenomena like pronunciations of individual sounds across languages or the co-articulation of sounds in a sequence [AD10].

2.2 Tasks

LID can be addressed via several classification tasks. Traditionally the problem has been addressed as a *closed-set identification* task: A speech input has to be identified as being voiced by one language l out of an a priori given language set $\mathcal{L}$. Those languages are modeled via language-dependent models. This is a forced-choice architecture, which will always identify l as one language out of the closed set, even if $l \notin \mathcal{L}$. Another setup in LID is the *open-set detection* or *language verification* task: It is to decide whether or not a voiced utterance belongs to a given target language L, resulting in a YES/NO answer. A Universal Background Model (UBM) called $\bar{L}$ is supposed to be complementary to L and is trained to represent a speaker- and language-independent distribution of features. The UBM training data should have the same general acoustic features than the expected input, i.e. type and quality of speech should be similar or equal, as well as the gender ratio [RQD00]. It is essential that all available languages flow into the model in equal shares [Rey09]. The most general *open-set identification* task can be solved by either parallelizing multiple verification systems or implementing a closed-set identification system followed by a back-end language verification module which produces a YES or NO answer for the winning language [Nav06].

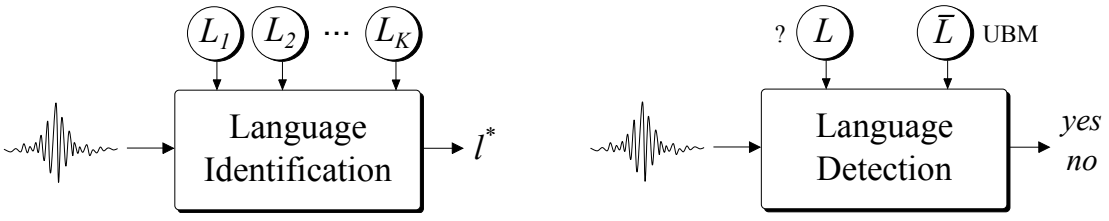

Figure 2.2: Schematics of the traditional LID classification tasks, closed-set identification (left) and open-set detection/verification (right) [AD10].

2.3 Performance Measurement

The quality of language identification tasks can be measured by means of identification accuracy. Therefor, an identification rate, averaged over all target languages is calculated. For a given language l the language identification accuracy is defined as

$$ACC_l = 100 \cdot \frac{s_{hit}}{|S_l|} \tag{2.1}$$

where s_{hit} is the amount of correctly identified test sequences of all test sequences S_l voiced in language l.

The NIST Language Recognition Evaluation[1] is very popular for evaluating LID performance. [NIS09] considers the language identification task a detection task: Given an utterance and a language hypothesis, the task is to decide whether or not that target language was in fact spoken. Therefor a yes/no decision for each target/non-target language pair is necessary. The performance of a system is characterized by its false rejection rate and false acceptance rate. For each detection task the output comprises: the yes or no decision, if the target language was spoken, and a score, describing the confidence of the decision. For each target/non-target language pair the performance is computed separately in terms of detection miss and false alarm probabilities:

$$C_{miss}(l_t) = C_{miss} \cdot P_{target} \cdot P_{miss}(l_t) \tag{2.2a}$$
$$C_{fa}(l_t, l_n) = C_{fa} \cdot P_{non-target} \cdot P_{fa}(l_t, l_n) \tag{2.2b}$$

where $\{l_t, l_n\} \in \mathcal{L}$ are target and non-target language respectively, C_{miss} and C_{fa} represent the relative costs of miss and false alarm events, P_{target} and $P_{non-target}$ are the a priori probabilities of the target and non-target languages. On this basis a cost performance averaged over all language pairs will be computed:

$$C_{DET} = \frac{1}{|\mathcal{L}|} \cdot \sum_{l_t} \left\{ C_{miss}(l_t) + \sum_{l_n} C_{fa}(l_t, l_n) \right\} \tag{2.3}$$

The results are visualizable in a Detection Error Tradeoff (DET) curve, whereas C_{DET} will be evaluated for every point along the curve by modifying the score thresholds to calculate the point with minimum cost [MDK+97]. An alternative visualization is the Receiver Operator Characteristics (ROC) curve. A comparable performance measure can in both cases be given as Equal Error Rate (EER), equalizing the contribution of false alarm and miss rates to the identification error [AD10].

2.4 Probabilistic Formulation

The main approach to automatic LID is very similar to state-of-the-art-approaches to speech- and speaker-recognition, which use a statistical framework. The core of the most common approach is the Bayes classifier. With help of mathematical formulation it is possible to decompose the task into several sub-problems. Identifying a language l^* upon a pool $\mathcal{L}$ of target languages can be formulated and transformed by the Bayes formula as follows:

$$l^* = \underset{l \in \mathcal{L}}{\operatorname{argmax}}\, P(l|X) = \underset{l \in \mathcal{L}}{\operatorname{argmax}}\, \frac{P(X|l) \cdot P(l)}{P(X)} = \underset{l \in \mathcal{L}}{\operatorname{argmax}}\, P(X|l) \tag{2.4}$$

which models the probability of X being observed when l is the spoken language. X is the acoustic observation according to the processed signal, $P(l|X)$ is the probability of l being observed, given X. $P(X)$ is the a priori probability of observing X. It is constant for classification decision and thus negligible. Assuming all languages to be equally probable we can further simplify by discarding $P(l)$.
According to this formula, the language is considered as a whole. Further decomposition reveals the components generating the overall probability. [Haz93] defines the acoustic

[1] http://www.nist.gov/itl/iad/

observation X as a compound of acoustic information v, F0 information f, and the most probable phonetic sequence a^* and corresponding segmentation sequence s^*. Here $v = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m\}$ and $f = \{\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_m\}$ are sequences of m vectors which represent the acoustic information and voicing information respectively. Using standard probability theory, we can exemplarily decompose equation (2.4) as follows:

$$l^* = \underset{l \in \mathcal{L}}{\operatorname{argmax}}\, P(l|a^*, s^*, v, f) = \underset{l \in \mathcal{L}}{\operatorname{argmax}}\, P(a^*|l) \cdot P(v|a^*, s^*, f, l) \cdot P(s^*, f|a^*, l) \tag{2.5}$$

Whereby (s^*, f) represents the prosodic information, a the phonemic information. The single components of equation (2.5) correspond to the basic speech components, which are:

$P(a^*|l)$
: The phonotactic model, describing the frequency of phone co-occurrences. This probability will further be decomposed into n-grams.

$P(v|a^*, s^*, f, l)$
: The acoustic model. Leaving a out, the model will be independent of phonemic information, leaving out (s, f), it will be independent of prosodic information. Leaving out l, the acoustic model will be language independent. In this case, a needs to be language independent as well.

$P(s^*, f|a^*, l)$
: The prosodic model.

2.5 LID Modeling

State-of-the-art LID systems in principal incorporate the same major components. Figure 2.3 depicts a diagram of a generalized LID system architecture forming a source-channel model. An acoustic front-end extracts appropriate feature vectors. This preprocessing and feature extraction is very similar to those of other language processing tasks like language recognition and make use of the same acoustic features. The Language scoring block is the main component and comprises a speech decoding and language decoding block. The former generates the speech unit sequence which is composed of symbols belonging to a certain sound inventory, i.e. phones. The language decoding block is responsible for scoring these sequences over a set $\mathcal{L}$ of target languages. The back-end module is a specialized linear or non-linear classifier which optimizes the final output, as in the case of a lattice based system, where multiple scores have to be normalized or combined in an appropriate fashion [Nav06]. The classifier makes a final decision based on the available scores. Each particular module performance has an influence on the final classification power.

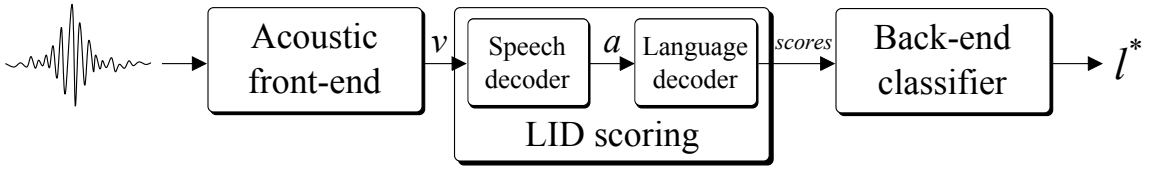

Figure 2.3: Schematic of the general LID system architecture, comprising the main modules acoustic front-end, LID scoring (comprising the speech decoder and language decoder) and back-end classifier.

2.6 Approaches

The language scoring block knows a large variety of implementations. There exist several architectures for every level of information, as well as fusion systems combining sources of information. It is self-explanatory, that each system design has it's strengths and weaknesses and it strongly depends on the field of application which approach will perform best.

2.6.1 Acoustic

Purely acoustic LID approaches want to capture the essential differences across languages by modeling the distributions of spectral features [Nav06]. Thus, the differences in short time acoustics across languages is used. Usually a language independent set of spectral features will be extracted from a set of training utterances. A statistical classifier is used to identify the language-dependent patterns in these features. There is no mapping to explicit linguistic units such as phones. Except the language-specific audio data, no language-specific knowledge in form of transcriptions is needed for training, making this type of architectures easily expandable to further target languages [AD10].
Early approaches were developed on the basis of Linear Predictive Coding (LPC) analysis and performed a simple spectral pattern matching [AD10]. An improvement was the use of Vector Quantization (VQ) for creating an acoustic code-book with more than just one reference pattern per language. Today most commonly used are Gaussian Mixture Model (GMM), Artificial Neural Network (ANN) and Support Vector Machine (SVM) approaches.
For the GMM approach, a GMM model is estimated for each language. The core principle is: an acoustic feature vector is assumed to be generated by a multivariate stochastic source with a probability density function [Nav06]. GMMs don't represent temporal dependencies very well, but with extended feature sets such as the Shifted Delta Cepstra (SDC), representing longer time spans, some compensation for this disadvantage is possible [AD10]. ANNs were introduced as an alternative. One strategy is to train an ANN for mapping feature vectors to one of several broad phonetic classes. These classes are then used for further LID. The disadvantage of this technique is the relatively high amount of required training data [Nav06]. An SVM system transforms the feature vector input into a high-dimensional feature space, in a way that a linear classification on averaged vectors can be peformed. The extension to multi-class classification can be done with several two-class SVMs in parallel [AD10].

2.6.2 Phonetic (PPR)

Systems using phonetic information require a certain amount of language-specific knowledge, in contrast to the acoustic approaches. Required are phone inventories and labeled acoustic training data. The assumption is, that different languages use different sound inventories. With phone-based Hidden Markov Models (HMMs), sound units such as phonemes can be modeled. HMM-based models most commonly consist of three states, forming a feed-forward, left-to-right topology, giving HMMs improved temporal modeling capacities. Each state comprises different GMMs for the modeling of specific sounds.
For each target language a separate phone recognizer is used. For that reason a phone inventory has to be defined and a set of acoustic phone models is needs to be trained. Language models, generally bigrams or trigrams, are estimated for modeling phone co-occurrences, i.e. phonotactic constraints, which will be applied during the decoding process of every single phone recognizer. Thus, the scores of PPR systems incorporate acoustic, phonetic and phonotactic informations. A back-end classifier determines the identity of the spoken language by comparing the recognizer scores against each other. A disadvantage

of the PPR approach is the required amount of language-specific knowledge for training, making it difficult to extend the systems to minority languages or languages lacking enough training material.

2.6.3 Phonotactic

The language-specific probabilities of sound unit occurrences and co-occurrences are one of the most widely used cues for language discrimination in state-of-the-art LID systems. Phonotactic LID is built on a probabilistic framework and uses the strong language-specific dependencies between individual phones, as well as individual frequencies of phone observations. The phonotactic constraints are modeled by n-gram phone based language models, which are, as for the PPR approach, most of the time bigrams or trigrams. Given an example phone sequence resembling the word "this", and n target languages a typical observation probability would be calculated as:

$$P(/DIs/|l_i) = P(/s/|/D/, /I/, l_i) \cdot P(/I/|/D/, l_i) \cdot P(/D/|l_i), i \in 1, \ldots, n \tag{2.6}$$

The inequality $P(/DIs/|l_i) > P(/DIs/|l_j), i \neq j$ should be true for $/DIs/$ occurring most frequently in language l_i.
The difference to the PPR approach is, that the acoustic decoding component no longer incorporates the language models [Zis96, AD10]. The decoder now serves as a mere speech tokenizer, which produces a discrete symbol sequence without applying language-specific phonotactic constraints.

2.6.3.1 PRLM

In the PRLM approach each language-specific n-gram phonotactic model is fed with the sequence of tokens that was generated by a language independent or multilingual phone decoder. The phone decoder or tokenizer is a research issue for itself, because finding an universal symbol inventory, which covers most of the phones occurring in the majority of languages, especially the target languages of a proposed PRLM system, is not trivial. Used in today's systems are broad phonetic classes, shared among different languages, IPA-like[2] inventories, or the union of several language-dependent inventories. A general rule is, that the performance increases with increasing size of the set, provided there is enough training material available [AD10].

2.6.3.2 PPRLM

The latter approach can be improved by running several PRLM systems in parallel, leading to the PPRLM approach. The parallel decoders may for example each be a language-dependent tokenizer, generating multiple streams of phones which are each scored by a separate set of phone set-dependent language models, one for every target language. The resulting scores will be normalized and combined for every language.
Just as for PRLM, the parallelized version requires language-specific labeled resources only for the acoustic model training for the speech tokenizers. A larger number of target languages may be covered simply with the training of further language models and language-scoring back-ends. Additional language-specific audio data for the training is the only resource needed.

[2] http://www.langsci.ucl.ac.uk/ipa/ipachart.html

2.6.4 Prosodic

Different languages are also characterized by differing intonation and rhythm patterns. Measurable are the fundamental frequency (F0) variations, energy and duration. The difficulty in using prosodic cues for LID is the large number of additional factors influencing F0 and amplitude, such as: speaker characteristics (voice, speaking rate and emotions), lexical choice (differing lexical tone and word characteristics) and content of the utterance (being a question, statement, emphasis, etc.). One of the open challenges is, how to separate irrelevant characteristics from the desired information. Purely prosody-based LID systems are rare. One of the recent studies made use of the temporal behavior of prosodic cues to segment and label test utterances into sets of discrete units, that represent 10 different prosodic classes, for example describing segment duration, falling and rising F0 and the energy. The evaluations demonstrated that prosody dynamics contain utilizable language-dependent information [AD10].

2.6.5 LVCSR

The approaches closest to the situation of human listeners being familiar with several languages are the ones that use knowledge about lexical and grammatical structure. Systems of that category use fully fledged Large Vocabulary Continuous Speech Recognizers (LVCSR) [Nav06]. Such a system decodes an utterance into a sequence of words and makes an analysis for language-specific patterns. The basic principle can be called a "brute-force" approach: an utterance in an unknown language is processed by all available LVCSR systems for all potential target languages to choose from [Nav06]. The language, whose speech recognizer system yields the highest score wins.
A huge advantage of this approach is, that the results are more accurate and less noisy than those of the low-level approaches. This is due to the application of high-level constraints, modeled by a lexicon and a language model. On the other hand, the complexity is a disadvantage because of the huge amount of required training data, which is only given for the most commonly spoken languages. Further, the computation is much more complex and requires more resources, which may pose a problem for hand-held devices etc.

2.6.6 Fusion

Today's research focuses on the combination of several systems to optimize the accuracy of LID. Especially the fusion of heterogeneous systems makes it possible to cope with the individual limitations of single systems. Most common fusion implementations combine the outputs of the individual components such as acoustic, phonotactic and prosodic scoring modules. A higher-level back-end classifier combines the scores in an appropriate way to make a final hypothesis. The fusion systems make their decisions based on multiple partly redundant information levels and thus imitate the human LID process.
Prosodic modeling was not performing very well as stand-alone systems. But as extension, i.e. to phone-based approaches, the performance was enhanced by exploiting prosodic features for example on syllable-level [AD10]. Another option is to incorporate lexical information for LID systems, for instance as a keyword spotting module, without fully modeling the language with a language model. This may be an option in situations, where the choice of words can be predicted in a way. An example might be flight scheduling services or comparable applications.
Instead of parallel fusion a conceivable alternative is a hierarchical structure. A novel framework called Hierarchical LID clusters language hypotheses in a way to form a tree-like structure [YAC08]. Each leaf is an individual language hypothesis, and the nodes are groups of languages with similar classifications. Thus, an utterance is classified level by

level, in a way, that at each level the most discriminative feature is used. Phonotactic and LVCSR components may also be combined, so that the phonotactic component performs a pre-classification and a smaller set of LVCSR systems undertakes a more fine-grained analysis [Nav06].

2.7 Design decisions

Depending on the purpose of a system, there are various design decisions to make. Trade-offs especially exist between the utilized level of information and

- the accuracy and robustness
- the minimum test segment length
- the duration of computations
- the amount of required training data.

The size of the language pool to choose a target language from, and whether or not rejection classes are used may also cause huge differences in identification accuracy and system complexity [YAC08].

2.7.1 Minimal Test Segment Length

One of the most important factors for the evaluation of LID systems, especially in real-time application environments, is the required time for a satisfactorily accurate average identification performance. The tolerable latency depends on the field of application. A conceivable real-time demanding application is a simultaneous speech translation system, where the knowledge about the spoken language of a user is not given a priori. That might as well be the case in multilingual telephone information systems, such as flight booking services. On the other hand, a low error rate could be much more of interest when the costs of errors in identification is very high. This may be the case in offline translation systems or speech-to-text dictation applications. It may be costly to restart the process, when the language in the first run was mis-identified.
Experiments in human LID have shown, that there is a positive correlation between test signal duration, language knowledge of the listener and the classification accuracy. The more languages a listener is aware of and the more comprehensive the language-specific knowledge is, the better is his performance, even on very short utterances [SMB96, LM95]. When asked, which types of cues the subjects used to discriminate the languages, they gave answers like *typical sounds* in form of syllables or short words, such as “ich” in German language, increased occurrence of nasals, words often ending in a vowel, etc., as well as *prosodic features* like a “sing-sang-rhythm” in Vietnamese [MJC94, LM89, Nav06].
The phonotactic techniques like (P)PRLM or syntactic techniques like LVCSR-based systems need sufficiently long input sequences to gain enough statistically significant information to provide accurate results. Those systems perform best on utterances with lengths greater than 10 seconds and remain competitive on shorter test durations. Given even longer test sequences, the results of the high-level LVCSR approaches are the best compared to all other approaches [Nav06]. Systems working on a sub-phonetic level, meaning acoustic events, are able to make fair decisions in less than a handful of seconds. Acoustic systems make use of the fact, that languages have unique sounds (nasals etc.), which can be identified even in short segments, also the transitions of sounds (e.g. aspirated vs. un-aspirated), which are characteristic for languages, are extractable from short segments [GP03]. A solution to this task may be a system, that extracts multiple phonetic features from an utterance in parallel. The phonetic features are abstract classes related

to articulatory properties such as *voiced*, *nasal*, or *vowel*, which can then be grouped into subset of feature classes, e.g. a *place of articulation* class. During testing, the signal is passed through parallel feature recognizers, followed by feature n-grams which compute the probabilities of each feature stream given a certain language [KP01].

2.7.2 Comparison of Approaches

In terms of performance, LVCSR systems are the most effective architectures, presumably due to their heavy incorporation of high-level information. Their great disadvantages are the high computational cost, along with the requirement of a large amount of labeled training data. Given ready-to-use speech recognizers, it is relatively simple to expand an existing LVCSR-based LID system. However, this approach is ill-suited for the expansion to minority languages.
Considering the performance-to-cost ratio, the phonotactic approaches yield the best overall performance [Nav06]. Labeled training data is required only for the training of the phone recognizers. With the decoders up and running, it is possible to train n-gram language models for any requested target language. That makes them the first choice for situations involving the necessity to discriminate a larger number of languages, as well as the work on rare languages or languages without sufficiently large training pools.
The PPR approach yields fair identification results even on test segments of only several seconds in length, because acoustic, phonetic and phonotactic information levels jointly contribute to the classification process [AD10]. Also, acoustic approaches tend to be very promising for short-time language identification with solid performance in ranges under 5 seconds. However, their overall capabilities as stand-alone systems remain under those of phonetic, phonotactic or LVCSR systems. Therefore, acoustic as well as prosodic language cues serve as supporting sources of information in today's fusion systems [Nav06].
Table 2.1 gives a rough overview of the LID standard approaches, considering the diverse fields of application.

2.7.3 Conclusion

During the decision-making procedure, which approaches to implement within the scope of this thesis, we isolated the phonetic and phonotactic approaches as most promising, since purely acoustic systems do not seem to yield satisfactory identification results for our purposes, and LVSCR architectures lack the ability to perform well in real-time demanding scenarios. Thus, we had a closer look at the individual advantages and disadvantages of the proposed systems, according to literature:
PPR systems tend to deliver fast and solid classification results on sufficiently long inputs, whereas the (P)PRLM architectures are able to gain very good identification results even on shorter test durations. Despite the need for a higher amount of input data for robust classification, today's phonotactic systems outperform the approaches that were implemented in the first place to address the short-time identification task [Nav06, AD10].
A major disadvantage of the PPR approach is the limited expandability to further target languages, as there is an appropriate pool of data required for each new phone recognizer to affiliate. On the other hand, if already trained acoustic models and language models are at hand, e.g. used by full speech recognizers, it needs only a few steps to build a language identification system from scratch. For most of the major languages, there already exist various sets of reliable models to work with.
For the PRLM approach, it may not be easy to find an appropriate phone inventory for the front-end, in such a way that most of the phones occurring in the languages to discriminate are covered. On the other hand, the use of several phone tokenizers in parallel by PPRLM frameworks may considerably enhance the acquisition of information with the aid of complementary sound inventories representing a single input signal in multiple ways.

Approach	Test duration	Strengths & weaknesses	Identification accuracy	Exemplary fields of application
Acoustic	∼3 s	Fast computation, no labeled data needed, sub-standard standalone performance	Good (for small language pools)	Real-time scenarios, low cost
Phonetic	>10 s	Fast computation, simple framework, costly in training	High	System with large language pool
Phonotactic	>5 s	Good performance-to-cost ratio, no labeled data needed for LM training, easily expandable	High, very high for longer inputs	System with large language pool and/or rare languages
Prosodic	≫10 s	Tolerant towards channel mismatches, not suitable as standalone system	Low	Pre-classification module
Syntactic	>5 s	High-accuracy solution, high computational and training costs	Very high	Multilingual dialog/translation system

Table 2.1: Comparison of the LID standard approaches, under consideration of common test time duration, strengths and weaknesses, average identification accuracy and exemplary fields of application [Nav06].

It is another strength of the (P)PRLM approaches, that the training for new target languages is very easy, since only the n-gram models for the phonotactic constraints have to be computed upon automatically tokenized audio data.
Despite the disadvantages, the proposed approaches tend to fit our requirements. Those are in detail:

High overall performance
: According to the research community, PPR and (P)PRLM systems yield about equally competitive results on test signal lengths above 10 s [Zis96]. Only LVCSR architectures seem to perform slightly better, with the major drawback of high costs in terms of training data and computation.

Solid short-time identification results
: (P)PRLM seem to require test segments with at least 5 s of length for reliable identification results, PPR systems tend to require slightly more data. However, even on shorter test signals today's common phonotactic systems outperform acoustic approaches, assumedly due to the use of higher-level information.

Low computational costs
: PPR architectures undertake faster Viterbi path computations in the decoding process due to the incorporation of a language model. In terms of computation, the (P)PRLM approaches are generally more costly than PPR, because the speech decoding step is a mere tokenization without the application of phone sequence constraints. However, in consequence of the very small dictionary, containing only the phone inventory, the decoding complexity is much lower than those of full speech

recognition tasks, hence much faster than LVCSR-based LID systems.

Easy expandability

Both, the phonetic and phonotactic levels of information have high discrimination power even on larger language sets, compared to the lower-level approaches [Nav06]. The (P)PRLM architecture is predestined for the application in scenarios, where easy expandability to new target languages is crucial. Given several ready-to-use acoustic model and language model sets, the disadvantage of the PPR approach carries not much of a weight.

3. Phonetic and Phonotactic Language Identification

The approaches to LID using phonetic and phonotactic cues are among the most successfully applied stand-alone architectures today. Their high language discrimination power on the one hand and their relatively simple implementation on the other justify their success. Since the early nineties, the parallelization of multiple phone decoders is a very popular LID architecture design [AD10].

3.1 Linguistic Background

Linguists know several sub disciplines dealing with the role of sounds in a specific language, the structural characteristics of sounds and sound co-occurrences, as well as their acoustic characteristics and their articulation and perception by humans. The discipline of emphphonetics aims at an objective description of sound acoustics with a focus on linguistically relevant aspects [AD10]. The assumption is, that different languages make use of different sound inventories. Even when there are significant parts shared between certain inventories, the realization of sounds and their occurrences might be very diverse regarding a specific language pair. Those differences can especially be modeled by language-specific sets of phone based Hidden Markov Models (HMMs) as they are already used since decades in speech recognition systems. Typical phone inventories range from 20 to 60 symbols [AD10]. The inventory generation strongly depends on the corresponding language and its specific sound characteristics. The inventory size may for example significantly increase when tonal information or gemination effects are embodied in discrete phonemes [AD10]. The principles in finding appropriate symbol sets are precision, robustness, modularity and transferability, as explained in detail in [ST95].
The linguistic field of emphphonology knows various types of information which help to discriminate one language from another. One is the said language-specific sound inventory, namely the phone set itself. Although many sounds are shared among languages, the compositions of those language-dependent sets may be very distinctive for various language pairs. There even exist very special sounds which can by themselves be language discriminating, or at least indicating the class of a language, for example the click consonant found solely in southern and eastern African languages [NP03]. The International Phonetic Alphabet (IPA) is today's most commonly used alphabetic system of phonetic notation. It was developed by the International Phonetic Association as a standardized representation of the entirety of all distinctive sound characteristics, i.e. phonemes, intonation and segmentation specifics in the world's languages [Ass99]. Phonotactics is a branch of phonology

and deals with the rules that define all legal co-occurrences of phones, given a language. Phonotactics focuses on the functional and linguistically distinctive roles of sound representations in a language system [AD10]. Consequently, phonotactic constraints are highly language-specific. In order to utilize phonotactic constraints for automatic language identification they are preferably modeled by n-gram language models, being the most popular representational structure [Nav06]. The models are calculated by analyzing the phone co-occurrences in each language on an adequate amount of training data. The phonotactic constraints can be incorporated into an LID system in at least two different ways. The first variant is to use the phonotactic model during phone recognition by the speech decoding block. Thus, the phonotactic rules have a direct effect on the phone sequence generation. This way of incorporating the models leads to the parallel phone recognition (PPR) approach. The second way of application is as back-end analyzer for a generated sequence of phones, thus the phonotactics are part of the language decoding block. The latter is done in the phone recognition followed by Language modeling (PRLM) approach. The parallel phone recognition followed by language modeling (PPRLM) approach finally incorporates multiple PRLM systems running in parallel.

3.2 Parallel Phone Recognition (PPR)

The PPR approach utilizes several single-language phone recognition front-ends, which run in parallel. The phone recognizers make use of phonotactic constraints during the decoding process. The resulting streams of sound tokens are directly influenced by the applied phonotactic rules. It is required to find appropriate phone inventories and to train acoustic models for every language l in the set of potential target languages $\mathcal{L}$. Therefore it is crucial to have a sufficiently large pool of labeled training data for each language-specific phone recognizer. The labels could either be phonetic transcriptions or orthographic transcriptions along with a phone based pronunciation dictionary [Zis96]. For most of the world's major languages suitable labels are available in sufficient amounts in order to estimate accurate models which reliably represent the acoustic language characteristics. For that reason, the drawback of the PPR approach is the poor expandability to languages with only very sparse training data pools and minority languages, especially languages without a writing system. To date numerous cultures exist without a written language: In the rain forests of Brazil, Venezuela and Columbia, in the Sahelian zone in the jungle of Malaysia, in the mountain valleys of Papua New Guinea and the Australian outback, each comprising only several hundred to thousand speakers [Haa07]. The vast number of different languages without a writing system renders this issue not neglectable.
Language identification is performed by conducting a Viterbi decoding of a test utterance by each single language-specific phone recognizer. A back-end classifier compares the scores generated by the decoders, representing the likelihood of the Viterbi path by each module. The language modeled by the winning recognizer is the language $l^* \in \mathcal{L}$ most likely being spoken. The phone sequence generated by the winning recognizer is optimal by means of the combination of acoustics and phonotactics, which also implies improved phone based transcriptions of the test data [Zis96].
Several design decisions for implementations include, but are not restricted to:

Alternative sound units
: Sound inventories other than phones may be syllables or words.

Modeling sound context
: Acoustic models can either be context-independent or context-dependent. In the latter case, phone-based sound units are known as polyphones.

n-gram language model characteristics
: For the n-gram language models it is decidable which n to use. In most of the cases

$n \in \{2, 3\}$. It is possible to use higher order n-grams such as fourgrams, but it has been shown that trigrams a sufficient to appropriately model phonotactics and show performance gains of 30% - 40% in comparison to bigrams, whereas the benefits of fourgrams remain insignificant [AD10]. Further, data sparseness is a problem concerning the training of accurate higher-order language models. Regarding smoothing or discounting techniques, the commonly used standards seem to do well for the purpose of phonotactic model training [AD10].

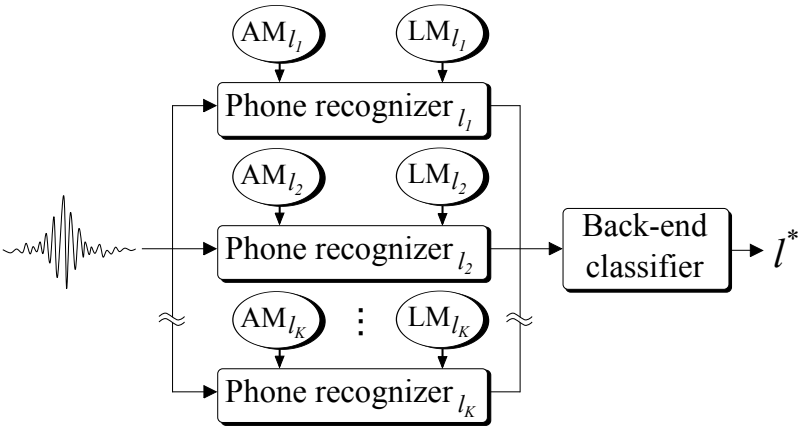

Figure 3.1: Schematic of a parallel phone recognition (PPR) system, which incorporates language models for language-dependent phone-based decoding and uses the recognizer scores for classification [AD10].

3.3 Phone Recognition Followed by Language Modeling (PRLM)

The PRLM approach uses only one single phone recognizer as decoding front-end. In contrast to the PPR system design the recognizer serves as a mere tokenizer. The decoding process produces a discrete symbol sequence without any language model influence. Instead, language models serve as back-end language scoring modules.
The symbol inventory of the phone tokenizer can be language independent or language dependent. The latter case implies several issues concerning the choice of appropriate symbols satisfying the language independency as well as the phone coverage, the robustness dependent on available training data and the overall precision. Another design decision is the size of the symbol set. Performance seems to improve with increasing size of the phone inventory, provided that the amount of training data is sufficiently large. Commonly used phone inventories include IPA-like phone sets or unions of several language-specific phone sets [AD10]. In case of the proposed use of a language-specific phone tokenizer, recognizers trained on English data seem to be the common choice because of the very good availability of accurate acoustic models or professionally labeled speech corpora for running a model training. Experiments revealed that single-language decoding followed by language model scoring yields solid identification results [ZS94].
The language decoding back-end consists of a bank of language-dependent phone based n-gram language models. For the training of the models, language-specific utterances are tokenized by the phone recognizer to be used in the LID system, resulting in a phone based training corpus for each future target language. From each corpus an n-gram language model is trained, resembling the specific phonotactics represented by the phones belonging to the inventory of the front-end tokenizer. Identification is performed as follows: During recognition the test utterance is tokenized and the phone sequence passes each language-specific n-gram model. A score for each model is generated, which represents the likelihood

of the utterance being voiced in the respective language. The language of the model with the highest score is selected as the language of the test message [Zis96].

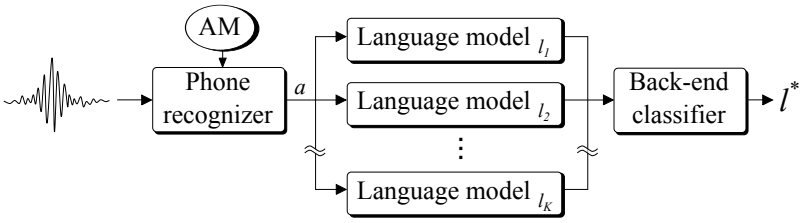

Figure 3.2: Schematic of a phone recognition followed by language modeling (PRLM) system, doing a phone-based tokenization and back-end language scoring, thus applying phonotactic constraints after decoding [AD10].

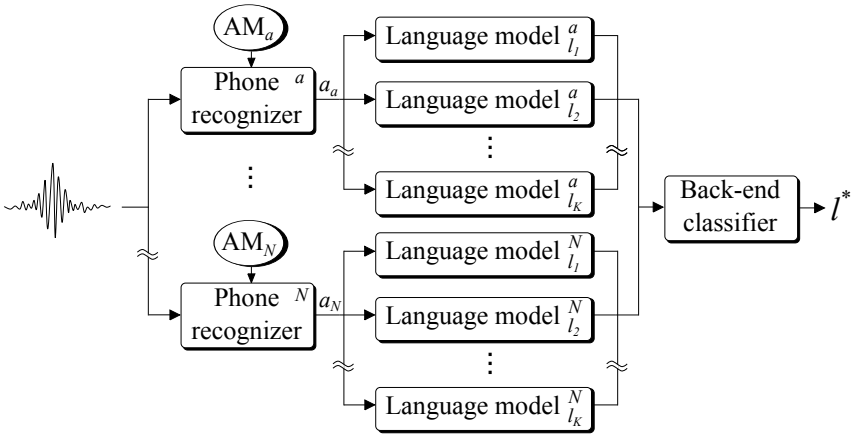

Figure 3.3: Schematic of a parallel phone recognition followed by language modeling (PPRLM) system, using several PRLM subsystems in parallel. Classification is done via averaged language model scores, combined in a language-wise fashion [AD10].

3.4 Parallel Phone Recognition Followed by Language Modeling (PPRLM)

The PPRLM approach makes use of several PRLM systems running in parallel for the multiple tokenization of an utterance. Following each tokenization, recognizer phone set dependent phonotactic models are applied, which produce likelihoods that the message was voiced in each of the respective languages. The models are trained for each target language and phone tokenizer pair. As for the PRLM approach, the phone sets of each tokenizer can either be language specific, or cover several multi language or multi phone category sets. It is even possible to additionally use other sound units, i.e. syllables. The assumption is, that the sounds in the target languages do not always occur in the inventory used to train the phone recognizer front-end [Zis96]. This approach gains an advantage by the higher coverage of recognizable phones or sound units in general. Generally, the PPRLM architectures tend to outperform PRLM systems, likely due to the improved robustness by using multiple sets of phonotactic models created for each of the tokenizers [Nav06].
As for the previous approach, a language model for language l is trained for each phone tokenizer r resulting in $l * r$ phone language models λ_l^r. This is done by decoding training data with each tokenizer and subsequent analysis of the phone occurrence and phone co-occurrence statistics. Another advantage for this approach is, that labeled training data

is only needed for the training of the phone tokenizers. Thus the expandability to other languages is very easily done by training additional language models per front-end for each new target language.
To perform language identification, an utterance is processed by all PRLM subsystems in parallel. The language model outputs will be appropriatly combined per language , e.g. averaged in the log domain [Zis96]. Classification is done on the fused overall language scores.

4. System Designs

We implemented two standard approaches using the levels of phonetic and phonotactic information as baseline systems for initial experiments. The first system is a PPR framework, which is evaluated against the second system, a PPRLM architecture. Our test scenario is a bi-lingual case, comprising the target languages German and English. During evaluation a special focus was on short test durations, because it is intended to use the results for the realization of LID front-ends for existing and future projects, which demand low latency identification performance.
We orient ourselves by the explications of [Zis96] and [AD10] for the baseline systems, except that we use an SVM back-end for output optimization and final language classification. The languages of the training data for the acoustic front-ends are the same as the target languages, that means we use German and English audio material for training the phone decoders. For the baseline systems, the SVM back-end classifiers do a classification on the specific features of each approach, that means acoustic scores for the PPR approach and language model scores for the PPRLM approach.

4.1 Test Database

Our test database, on which the experiments are run, is composed as follows: The German audio material mainly consists of recorded lectures, that were held at the Karlsruhe Institute of Technology (KIT) and the Carnegie Mellon University (CMU). The set is completed by several recordings of Landtag speeches as well as various talks of ceremonial acts. The English database is also comprised of lectures in large parts. It additionally contains audio material of TED Talks[1]. The total amount of recorded German data is 51 hours and the English data comprises 12 hours.
All of the data has been transcribed manually. Each transcription of a recording is organized in turns, whereas a new turn begins at the beginning of a new sentence or after a position in the audio signal which encompasses more than 300 ms of silence. All recordings are available in 16 kHz and 16 bit quality, most of them were done with close-talk microphones.
With the help of the transcription and turn boundary informations we generated five sets of audio segments per test language, differing in average segment length, namely 30s, 20s, 10s, 5s and 3s average. For the generation of the sets we used a partitioning algorithm, which iterates over every turn of a record and merges the current turn with a subsequent

[1] http://www.ted.com

turn, as long as various thresholds are not exceeded, where the constraints for considering a segment as legal are:

A maximum of 20% of silence within the segment, whereby only the silence between turns is taken into account. That means, the amount of pure speech within a segment may be significantly shorter than the overall segment length if there is a high amount of short parts of silence encompassed in the respective part of the audio signal.

A maximum of 20% of foreign words per segment, whereby the foreign words are identified by language tags in the transcriptions. Not all foreign words are tagged accurately, hence there may be more foreign words in a segment than the correctly identified foreign words.

A maximum of 20% variance in the segment length, compared to the intended average length. As long as the maximum length is not exceeded, new turns will be added to the currently processed segment. In case of an exeedance the current turn will be cut off. The following turn is the starting point for the next segment to determine.

Each database entry consists of a segment ID, a speaker ID, the segment length and time span in the original audio signal, an audio file path and file ID as well as the identity of the spoken language. For most of the segments, but not all, exists a phone-based transcription, which was generated by doing a Viterbi alignment of the existing word-based transcriptions. Because it was not possible to perform an alignment for all segments due to missing vocabulary entries, some remained untranscribed. The transcriptions were of importance for the performance evaluation of the phone recognizer front-ends, which will be discussed later.
The segments with 30 seconds of length are divided in a training set and a test set. The training set is used to train the SVM back-ends, which do the final language classification. The test set is used for the experimental tests. This split up is done in a way that the resulting sets of segments are speaker and recording disjunct. That means, that a quantity of segments originating from the same recording is not allowed to be distributed to more than one of the sets. Because the shorter segments of length 20s, 10s, 5s and 3s are generated upon the same recordings and thus comprise the same audio material than the segments of the 30s category, all recordings and speakers used in the training set for the SVM classifiers are dismissed. All test sets comprising the segments shorter than 30s use only the audio material which is also basis for the test segments of 30s length to guarantee that every test set is speaker and recording disjunct to the classifier training set.
Due to the vast amount of recorded data from one specific speaker, which in its entirety makes up to 47% of all available segments, we were forced to use the material of this speaker in both types of sets. Data of this specific speaker covers approximately 50% of the training set and on average 37% of each test set. However, the recording disjointedness is still maintained. Table 4.1 shows the amount of segments for each set of a specific length and purpose.

4.2 Feature Extraction

The acoustic front-ends of our proposed systems have been trained and tested with the help of the Janus Recognition Toolkit (JRTk), which features the IBIS single-pass decoder [SMFW01]. The feature extraction is done equally for each of our systems. The pre-processor is a variation of the ones used in [SKN11] by the KIT systems participating in the Quaero[2] 2010 speech-to-text evaluation campaign.
During training and decoding, spectral features are obtained every 10 ms by a Fourier

[2]http://www.quaero.org

		30s	20s	10s	5s	3s
Training Set	DE	1856	-	-	-	-
	EN	523	-	-	-	-
Test Sets	DE	1715	2436	3959	5224	5207
	EN	387	532	917	1325	1453

Table 4.1: Speaker-disjunct and recording-disjunct fragmentation of the database into a training set composed of the segments with lengths of 30 s, and test sets for each test duration category.

transformation using a Hamming window. Following the absolute value calculation a Mel filter bank reduces the spectral feature vectors into a 30 dimensional space. After logarithmic calculation an inverse cosine transform is applied, resulting in 30 Mel frequency cepstral coefficients (MFCCs), where the 13 lowest coefficients are selected as features. The mean and variance of the cepstral coefficients are normalized on a per-utterance basis [SKN11]. For the incorporation of temporal information seven adjacent frames are combined via vector stacking into one single feature vector. A linear discriminant analysis reduces the vectors to 42 dimensions and maximizes the discrimination of the classes.

4.3 Acoustic-phonetic Modeling

Our phone recognition front-ends employ left-to-right Hidden Markov Models (HMMs), which model each phone with three HMM states. Each state features a self loop and a transition to the next state. We use language-specific phone recognizers, incorporating acoustic models trained for the same languages covered by our set of potential target languages, German and Enlish. For each of the target languages a separate phone set is trained. The German set consists of 45 phones, the English of 52 phones, including noise tags and silence tags. All models are semi-continuous quintphones that use 16.000 distributions over 4.000 codebooks, trained by an incremental splitting of Gaussians training and subsequent Viterbi training [SKN11]. Each model has up to 128 Gaussians. We have used training material from multiple domains. The majority of the data is from the web, and another large percentage is covered by broadcast news data and European parliament plenary sessions (EPPS). Table 4.2 lists the sources of training data for both phoneme sets.

Source	Amount [h]	Source	Amount [h]
Tagesschau	17.5	EPPS	80
ISL database	16	TED talks	9.8
Globalphone	17	auto-transcribed EPPS	80
Verbmobil	57	HUB-4 corpus	140
Landtag BW	123		
KIT/CMU recordings	25		
Quaero training data	19.5		
Sum	275	Sum	309.8

Table 4.2: Data sources for training the acoustic models.

4.4 PPR System

The parallel phone recognition approach was the first strategy we implemented as a real system. Given fully fledged speech recognizers for German and English, which took part in the Quaero 2010 speech-to-text evaluation campaign, we were able to do a "warm start" by using the available resources. We re-trained the acoustic-phonetic models for each phone recognizer front-ends without vocal tract length normalization (VTLN), but the overall architectures remained the same. The word based dictionary of each decoder is replaced by a dictionary comprising the respective phone set, thus enabling a phone based tokenization of an audio input instead of a fully word based speech recognition. Further, the phone-based language models as an integral part of each phone recognizer in the PPR approach had to be trained. The SVM back-end classifier was trained with language cue feature vectors, which include the recognizer scores for each training utterance.

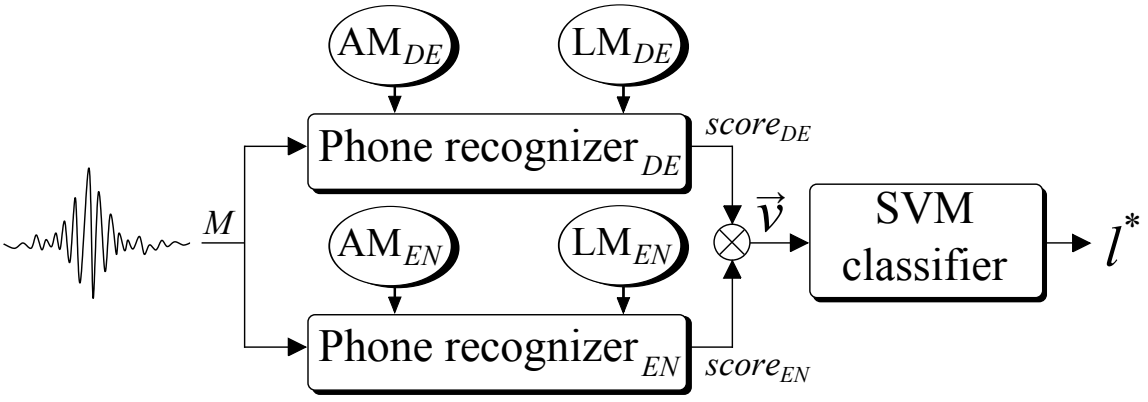

Figure 4.1: Schematic of our PPR system, using German and English phone recognizers. Classification is done by an SVM back-end classifier and the Janus-specific recognizer scores, stacked to a language cue vector $\vec{v}$.

4.4.1 Training Language Models

In the PPR approach, phonotactic knowledge is directly incorporated in the decoding process of a test utterance. Thus, phonotactic, as well as acoustic-phonetic knowledge equally contribute to a generated decoder score.
Phonotactic information is modeled by phone-based n-gram language models, generated upon large training corpora. We trained two language models for both decoder front-ends. For that purpose we read out the labels of the data used to evaluate the KIT systems of the Quaero evaluation campaign, as mentioned in 4.2. The phone sequences form phone-based and language-specific training corpora. The German corpus comprises 10.362 sentences with a total amount of 1.936.001 phones, the English corpus consists of 159.046 sentences and 11.840.864 phones respectively. We used the SRILM toolkit for the computation of all n-gram co-occurrences and the generation of the n-gram language models. Because of the sufficiently large amount of training data we decided to use trigrams. According to [AD10] all standard smoothing methods seem to do well for phonotactic modeling. However, experiments in [WAC06] showed, that the Witten-Bell discounting method works best given a LID system utilizing phonotactic constraints in a PPRLM architecture. We used the same technique in all our architectures.
The general idea behind the Witten-Bell discounting is to look at a zero-frequency n-gram as to an event that just did not happen yet [WB91]. If it will happen, than it will be the first time it is observed. Thus, the probability of the observation of a zero-frequency n-gram can be modeled with the probability to see this n-gram for the first time. This probability is calculated by counting the initial observations of all different n-grams. The n-gram

counts are conditioned on a history: In order to exemplarily calculate the probability of an unseen bigram "$w_x w_i$" the probability of seeing a new bigram accoring to the scheme "w_x *" is used, making the new estimates of zero-frequency bigrams specific to this history [JM09]. The total probability mass of all zero-frequency n-grams is defined as:

$$P(\textit{next n-gram is novel}) = \sum_{i:c(w_x w_i)=0} p^*(w_i|w_x) = \frac{T(w_x)}{N(w_x) + T(w_x)} \tag{4.1}$$

where $T(w_x)$ is the number of types, i.e. all distinct n-grams, occurred so far, $N(w_x)$ is the number of all seen n-grams, and $c(w_x w_i)$ is the n-gram count for n-gram $w_x w_i$. The parameters are conditioned to a history w_x. The new n-gram probabilities for the cases $c(w_x w_i) = 0$ (n-gram $w_x w_i$ was not seen in the training corpus) and $c(w_x w_i) > 0$ (n-gram $w_x w_i$ was observed at least once) respectively, are:

$$p^*(w_i|w_x) = \begin{cases} \dfrac{1}{|\mathcal{N}_0(w_x)|} \cdot \dfrac{T(w_x)}{N(w_x) + T(w_x)} & \text{if } c(w_x w_i) = 0 \\ \dfrac{c(w_x w_i)}{N(w_x) + T(w_x)} & \text{if } c(w_x w_i) > 0 \end{cases} \tag{4.2}$$

where $w_x w_i \in \mathcal{N}_0(w_x), \forall i : c(w_x w_i) = 0$. In order to gain the probability mass needed to generate non-zero probabilities for the zero-frequency n-grams, it is necessary to discount all observed n-grams in the way equation (4.2) shows. The smoothed probability of zero-frequency n-grams is higher, if more n-grams with history w_x were observed.
Table 4.3 shows the compositions of our phone-based n-gram language models used by the decoders.

	LM_{DE}	LM_{EN}
Corpus size (#phones)	1936001	11840864
1-grams	48	55
2-grams	1917	2420
3-grams	27858	55837

Table 4.3: Size of the training corpora and the n-gram counts for each n-gram language model applied during the decoding process of the respective phone recognizer.

4.4.2 Decoder Parameter Tuning

The PPR approach does a classification on the phone recognizer scores, which incorporate the acoustic score and the language model score in the same way. In order to achieve the most likely phone sequence as hypothesis to a test utterance during the decoding step, it is required to tune the parameters of each decoder which regulate the impact of the language model on the tokenization process. Each decoder should yield optimal recognition results for utterances voiced in the language the phone recognizer is trained for. The IBIS decoder used by JANUS scores the hypothesis related to an input utterance as follows:

$$P(W|X) = \frac{p(X|W) \cdot P(W)^{lz} \cdot lp^{|W|}}{p(X)} \tag{4.3}$$

The lz parameter defines the language model weight, i.e. determines the impact of the language model on the decoding process relative to the acoustic model. The parameter

lp is a hypothesis length penalty or more precisely a word transition penalty, helping to normalize the sequence lengths of words W [SMFW01]. In case of our phone recognizers this parameter constitutes a phone transition penalty, as the only words in the search dictionary are phones. The adjustment of the lz, lp value pair was done manually for each recognizer. Therefor we utilized the data pool presented in 4.1. In each case we used 2 hours of randomly selected data from the data pool presented in 4.1 for calibration. For performance measurement, a simple phone recognition rate (PRR) was calculated for each utterance hypothesis and reference pair:

$$PRR = 100 \cdot (1 - \frac{N_{sub} + N_{ins} + N_{del}}{N}) \tag{4.4}$$

where N is the total amount of phones in the reference. N_{sub}, N_{ins} and N_{del} are the amounts of phone substitutions, phone insertions and phone deletions respectively, counted during the matching of the hypothesis to the reference transcription. During testing, noise tags and silence tags were discarded. With a PRR of 57%, the German phone recognizer yields a noticeably better performance than the English, achieving 53%. This might be due to the smaller number of elements in the German phoneme set, hence the risk of confusion may be smaller. The IBIS decoder enables the use of lattice rescoring during the decoding process: The best hypotheses generated with different *lz/lp* values are read from the search graph. A subsequent modification of the language model weight for this specific graph leads to a new one-best hypothesis. Because of the neccessary re-calculation of probabilities within the graph, this method is called rescoring [SMFW01]. We decided to take this option into account for our experiments, dut to potential improvements in the overall system performance.

4.4.3 Back-end Classifier

The language classification is done by the analysis of language cue feature vectors: For each test input, a vector combining language-specific cues is generated upon the phone recognizer outputs. For our baseline system, a feature vector corresponding to an input consists of a phone recognizer score according to equation (4.3) for each of the decoder front-ends.
Our proposed system utilizes an SVM for the vector classification. The classifier is trained from all feature vectors $\boldsymbol{v}_i = \{l^i_{ID}, score^i_{DE}, score^i_{EN}\}$ each corresponding to a segment s_i of the training set. Both, SVM training and later classification runs are done with the tools of the LIBSVM library (see Section 1.4). Due to the bilingual task, a two-class SVM is sufficient, but with the utilized tools it is easily possible to extend the proposed classifier to a multi-class scenario. We decided to use the C-Support Vector Classification (C-SVC) formulation, as it is the original SVM formulation [CV95], and fits our requirements. Given a training set of vectors $\boldsymbol{x}_i, i = 1, \ldots, l$ and a class-assigning vector $\boldsymbol{y} \in R^l, \boldsymbol{y}_i \in \{-1, 1\}$, the solution of the following optimization problem is required:

$$\begin{aligned} \min_{\omega,b,\xi} \quad & \frac{1}{2}\boldsymbol{\omega}^T\boldsymbol{\omega} + C\sum_{i=1}^{l}\xi_i \\ & \boldsymbol{y}_i(\boldsymbol{\omega}^T\phi(\boldsymbol{x}_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, i = 1, \ldots, l \end{aligned} \tag{4.5}$$

where ϕ maps $\boldsymbol{x}_i$ into a high-dimensional space [CL11]. The SVM will find a hyperplane in this space, which separates the the classes in a linear fashion and with a maximal margin

between them. The soft margin parameter $C > 0$ is a penalty parameter of the error term. The decision function for classification is:

$$\operatorname{sgn}\left(\sum_{i=1}^{l} \boldsymbol{y}_i \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b\right) \tag{4.6}$$

where $K(\boldsymbol{x}_i, \boldsymbol{x}_j) \equiv \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j)$ is the kernel function [CL11]. According to test runs on a small set of randomly selected data, the linear kernel type function $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^T \boldsymbol{x}_j$ seems to work best for us. During classifier training, the linear penalty parameter is determined automatically via a grid-search. For evaluation purposes, the SVMs are trained for probability estimates.
We decided in favour of the SVM approach because of several advantages over other commonly used classifiers, such as the naive Bayes classifier or ANN:

Training complexity
The SVM training is extremely fast, provided a suitable choice of features, that hold good discrimination capabilities.

Model complexity
SVM models are represented in an easy fashion, due to the sole need of support vectors defining the classes and class boundaries.

Classification complexity
Because of the kernel trick, an actual calculatory transformation of the vectors into a high-dimensional space is not necessary, which makes classification very fast.

Generalization
SVMs are maximum-margin-classifiers, hence their architecture aim at optimal generalization by establishing a margin of maximal width between the classes to discriminate.

Linear separability
The kernel trick enables linear separability of non-linear problems in a high-dimensional space. With adjustment of the linear penalty function parameter C, further flexibility is provided by allowing a certain amount of penalized misclassifications.

Bias compensation
During training, SVMs implicitly compensate for biased features: Scores of different recognizer systems usually are not comparable directly. Differences in the structures of underlying models, as well as the training and test data, cause biased scores in a way that scores of a recognizer r are higher, on average, than scores from another recognizer s [Zis96].

4.4.4 Performing Language Identification

Language identification with our PPR system is performed by doing a Viterbi decoding of a test utterance for each single language-specific phone recognizer. Each recognizer calculates the most likely path through the search graph, influenced by phonotactic constraints in form of phone-based n-gram language models. A log-likelihood score for the test utterance, given the modeled language, is generated by each decoder. The scores are normalized by utterance frame count and combined to a language cue feature vector for SVM classification. The SVM back-end classifier makes the language identity decision and outputs the hypothesis $l^* \in \mathcal{L} = \{\text{DE, EN}\}$ given the test utterance.

4.5 PPRLM System

The phone recognizers of our proposed PPRLM system use exactly the same models as the PPR architecture. The difference is, that the decoders no longer incorporate language models for the application of phonotactic constraints during the decoding process. Instead, phonotactics are modeled by language model back-ends. We implement the PPRLM system by training two PRLM systems: For both phone recognizers each, back-end language models for our two target languages are trained. This results in two times two phone set dependent and language-specific language models. During testing, an utterance is processed by each of the PRLM sub-systems in the same way: The test signal is tokenized by the phone recognizer. This token-dependent symbol stream is analyzed by the back-end n-gram models, which results in a language model score for each target language.
After PRLM processing, the generated scores are combined language-wise, and stacked to a feature vector. An SVM back-end classifier generates the hypothesis for the language being spoken. Besides the use of differing features, the SVM is trained in the same way as for the PPR system.

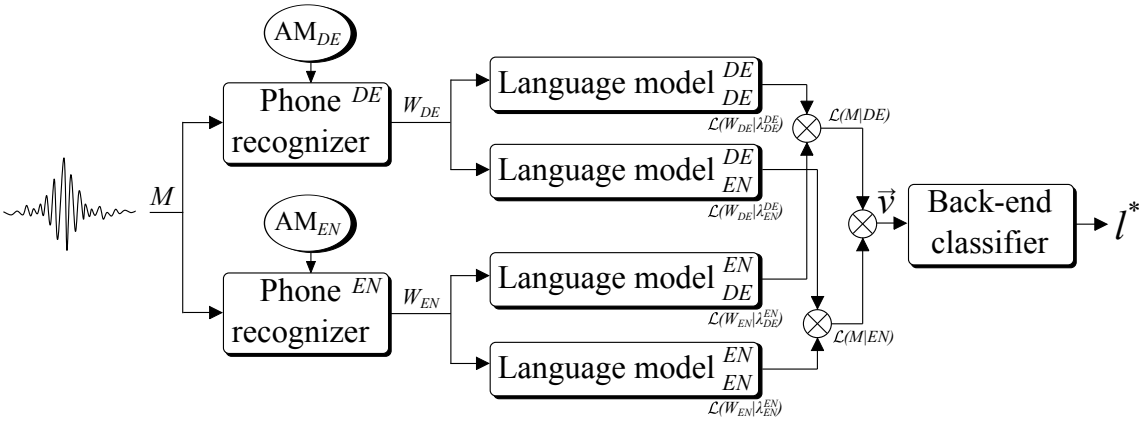

Figure 4.2: Schematic of our PPRLM system, using German and English phone tokenizers. Classification is done by an SVM back-end classifier and the combined and averaged language model sentence scores, stacked to a language cue vector $\vec{v}$.

4.5.1 Training Language Models

In order to perform a pure tokenization of the input messages, the language model weight parameter of both phone recognizers is set to $lz = 0$, excluding any language model influence during the decoding process. With the length penalty parameter set to $lp = 0$, the average hypothesis length to average reference length mismatch is minimal for both decoders. Lp values differing from 0 resulted in a higher mismatch of the average lengths, tested on the sets we also used in 4.4.2.
We ran the training of the back-end language models separately for each of our phone recognizer front-ends using audio data of the KIT 2010 Quaero evaluation campaign, resulting in two times two phone set dependent and language-specific phone-based n-gram language models. By tokenizing language-specific German and English data we get a phone-based corpus each, which ought to cover the phonotactic characteristics of the respective language. Both corpora are dependent on the specific tokenizer phone set. We calculated the language models upon the phone co-occurrence statistics of tokenized data with the SRILM tools, using the same settings as for the previous system.

Hence, each target language is modeled with it's natural phone set and the phone set of the opposing target language. The language scoring module has the following structure: The German phone tokenizer is followed by a German and an English language model,

	λ_{DE}^{DE}	λ_{DE}^{EN}	λ_{EN}^{DE}	λ_{EN}^{EN}
Corpus size (#phones)	2217564	20999643	2054232	10114569
1-grams	48	48	55	55
2-grams	2012	2021	2695	2700
3-grams	53580	79027	71362	104185

Table 4.4: Size of the training corpora and the n-gram counts for each n-gram language model applied as back-end language scoring modules for the PPRLM system.

both modeled upon the German phone set. The English phone tokenizer is followed by language models of the same languages, but modeled upon the English phone set. Thus, an utterance is represented by two streams of phone symbols instead of just one, as for the simple PRLM approach, which implies more information for the purpose of language discrimination.

4.5.2 Back-end Classifier

Again, the SVM back-end classifier is trained with the training set of our database and LIBSVM, with exactly the same settings as for the PPR language models. Each training segment is decoded by the phone tokenizers, resulting in two separate streams of symbols. Both streams are analyzed by the respective n-gram model back-ends: Let $\mathcal{T} = \{\text{DE, EN}\}$ be the set of target languages, $W_r = \{w_1^r, w_2^r, \ldots, w_m^r\}$ with $r \in \mathcal{T}$ be the phone sequence produced by the decoder front-end using the phone set of language r, and be $l \in \mathcal{T}$, then

$$\mathcal{L}(W_r|\lambda_l^r) = \sum_{i=2}^{m} \log P(w_i|w_{i-1}, \lambda_l^r) \tag{4.7}$$

is the log probability, that W_r is produced by the language model λ_l^r for target language l, trained upon the phone set r. Thus, the German phone set dependent language models generate scores for both modeled target languages each, given the stream consisting of German phones, with the procedure being the same for the English phone stream. Both scores of a particular target language l are averaged in the log domain, as both decoders are seen as working independently of each other, resulting in a joint language score $\mathcal{L}(M|l)$ for a given test message M, incorporating the scoring informations of both PRLM subsystems:

$$\mathcal{L}(M|l) = \frac{\sum_{r\in\mathcal{T}} \mathcal{L}(W_r|\lambda_l^r)}{f} \tag{4.8}$$

where f is the amount of used phone recognizers, i.e. the number PRLM sub-systems. In our case is $f = |\mathcal{T}|$.
A feature vector which represents a test utterance for SVM processing comprises the ID of the language actually being spoken, and the previously generated language scores for each of the target languages.

4.5.3 Performing Language Identification

For language identification, a Viterbi decoding of a test utterance is done by each single language-dependent phone tokenizer. Attention should be paid to the fact, that the decoding is done without language model influence, in contrast to the decoding in PPR architecture. For each resulting phone stream W_r, log-likelihood sentence scores are calculated for every language model λ_l^r. The scores are combined and averaged language-wise,

followed by a length normalization according to the utterance frame count. Finally, the scores are stacked to a language cue feature vector for SVM classification. The SVM back-end classifier generates a hypothesis $l^* \in \mathcal{T}$, given a feature vector.

5. Experimental Results

The two baseline systems were compared by performing bilingual, forced-choice classification experiments on sets of test messages with various average lengths. We tested on the segment lengths 30s, 20, 10s, 5s and 3s. Our focus was on the system's performances on the very short segment category, as the systems are evaluated with the aim to gain knowledge about the expectable performance in (near) real-time language identification scenarios. According to the conclusions of [Zis96], it was to expect, that the PPR and PPRLM approaches in general perform about equally.
The results for the baseline system experiments are shown in table 5.2. As can be seen, the PPRLM system outperforms PPR, regardless of the segment length category. This is not surprising, as PPRLM systems are able to gain more information from an input signal due to the use of multiple phone sets for message decoding. The PPR system solely relies on the phone recognizer scores, which are generated by each front-end separately during decoding. The information displayed by these scores apparently are of less quality for appropriate language discrimination, at least for the observed message lengths. However, with the test messages getting shorter, PPR performance decreases less fast, compared to the PPRLM system. This may lead to the assumption, that the phone recognizer scores, which incorporate acoustic as well as phonotactic valuation, are more robust, compared to the phonotactic scores delivered by language model back-ends, as they need a sufficient amount of statistically relevant data for proper probability estimation.

	DE		EN	
	PRR	lz/lp	PRR	lz/lp
PPR	57.1	28/-22	53.1	27/-16
PPRLM	46.9	0/0	40.9	0/0

Table 5.1: Phone recognition rates and lz/lp values of the front-ends used in our PPR and PPRLM frameworks respectively.

During parameter optimization for the recognizer front-end, we measured the phone recognition accuracy on the native language for each recognizer. As measurement we used the phone recognition rate (PRR), as explicated in equation (4.4). The results are shown in table 5.1. As mentioned before, we tried to take advantage of lattice rescoring, especially for the PPR phone recognizers, in order to potentially enhance the classification performance of the SVM back-end. Experiments on our database showed, that the use of lattice

rescoring did not affect the phone recognition performance at all. As a result we discarded the application of lattice rescoring during the phone recognition process.
With the baseline PPRLM system already working very well by utilizing the language model scores as language cues, we took the opportunity to test the language discrimination capabilities of the perplexity (PPL) measurement, which can be easily computed with help of the SRILM toolkit we used for language model training. The PPL of a language model λ_l for language l, given a test example message $W = w_1, w_2, \ldots, w_N$ is defined as:

$$PPL(\lambda_l, W) = 2^{\sum_{i=1}^{N} \frac{1}{N} \log P(w_i|w_1,\ldots,w_{i-1},\lambda_l)} \tag{5.1}$$

The idea is to evaluate, how well a language model may predict a test message, which for the test case is assumed to originate from λ_l. Thus, the perplexity makes a statement as to what extent a given language model is qualified to predict events presumably belonging to this model. A language model λ^* will assign a higher probability to a test message W, if it is more suitable to describe it. Thus, the PPL will be lower for λ^*, since the model is "less surprised" by this specific sample [Sch99]. The language used for voicing a specific message may be classified as the language of the model with the lowest PPL, as it seems to be most suitable to describe the message.
For each target language of our system, an averaged perplexity is computed:

$$PPL_l = \frac{\sum_{r \in \mathcal{T}} PPL_{\lambda_l^r}}{|\mathcal{T}|} \tag{5.2}$$

where $PPL_{\lambda_l^r}$ is the perplexity of the language model for $l \in \mathcal{T}$ corresponding to the phone set of the language-dependent phone recognizer r. As reminder, in our system each phone tokenizer corresponds to one target language, i.e. it is $r \in \mathcal{T}$. The averaging by $|\mathcal{T}|$ must be viewed as the averaging over the amount of phone tokenizer front-ends, or PRLM sub-systems, rather than the amount of target languages.
Using the averaged language model perplexities as features for language discrimination results in a slight deterioration of identification accuracy on very short segments. However, we made a noteworthy observation concerning the SVM model complexity: Surprisingly, utilizing the language model scores resulted in a SVM model consisting of an amount of support vectors 40 times higher than the quantity of vectors for the classification by PPL based features. One reason might be, that the transformation (5.1) of the language model scores in the form of equation (4.7) is more suitable for the discrimination by our SVM, already given a proper kernel function and optimized parameters. With that in mind, we observed both feature types as alternative approaches throughout further experiments.

System	30s	20s	10s	5s	3s
PPR	91.1	90.9	89.4	87.7	86.5
PPRLM (LM scoring)	99.7	99.7	98.8	96.0	92.0
PPRLM (PPL scoring)	99.8	99.6	98.9	95.7	91.1

Table 5.2: Identification accuracy (in %) of the PPR and PPRLM baseline systems and each test segment length category. Tests were done for the language model sentence scoring and perplexity measure scoring respectively.

5.1 Experiments on System Improvements

The purpose of further experiments was to evaluate potential performance gains by additional features such as scores of keyword spotting techniques, phone sequence statistics, multiple language model types, as well as specific feature combinations.

5.1.1 Experiments on the PPR System

Initial attempts to boost the PPR system included to test alternative scores to the score generated by the phone recognizers: A score generated by a JRTk-based recognizer system comprises several informations according to the hypothesis for a processed signal, namely an acoustic model score, the contribution of the language model, a word penalty (in this case a phone penalty), normalizing the amount of elements in the hypotheses, and a filler word/phone penalty. We isolated the pure acoustic score and ran tests using this score as sole feature for language discrimination. With purely acoustic scores, the performance dropped by 3% relative for 30s segments to 9% relative for the 3s category. The experimental combination of both, the unmodified recognizer scores plus the purely acoustic scores as additional feature, caused an improvement of 7% relative for the longest test samples, whereas the tests on short segments experienced a performance drop of 2% relative. That leads to the assumption, that a multiplication of acoustic information increases the discrimination capabilities, given sufficiently long input messages, whereas the performance on short segments suffers from lack of enough significant data. We decided to hold on to the unmodified scores, as the performance of the original setting seems to fit best for short-time testing.
Another attempt to increase the discrimination capabilities of the SVM classifier was to incorporate statistical knowledge regarding the phone sequence hypotheses. We tried to utilize the average hypothesis length, according to the amount of phones, and the average phone duration length, as this might display characteristics of the decoder's segmentation procedure. However, none of the additional features had a positive effect on the system's performance. Assumedly, the informations added in this way are too general.

5.1.1.1 Keyword Spotting

We tested on the incorporation of language-specific lexical knowledge into the PPR framework. The assumption is, that a test message W_l in language l would generate more phone sequences resembling common words of language l, than a test message in a language $l' \neq l$. We generated lists of the 100 most common words per language, using existent word frequency lists, which are computed upon corpora of transcribed TV and movie recordings [Wik11]. The corpora comprise 25 million words for German and 29 million words for English. For the 100 most common words per language, the pronunciations, including pronunciation alternatives were automatically extracted from large dictionary files. Finally, the lists were checked and modified manually, e.g., by deleting words or pronunciation alternatives consisting of single phones, such as the English "a".
We tested on two variants of application, the first approach is the use of generic n-gram language models, which are computed upon the word lists and the corresponding phone set, where the pronunciations of the keywords have a very high count, being the same for each word, and the phone set slips in with single phone counts of 1. The generic language models are applied as back-ends for the phone recognizers. The expansion of the baseline architecture resulted in a performance gain for all but the shortest test segment category. Furthermore, the classification accuracy became unbalanced, according to the per-language error rate. A more straight-forward approach of keyword spotting yielded more promising results. A phone sequence parser checks a recognizer hypothesis for sequences listed in the keyword pronounciation dictionary. The dictionary search runs sorted by phone sequence length, i.e. a phone already associated to a keyword cannot be associated again to a second keyword. Phones assumedly belonging to a keyword are marked as such. Best language discrimination results were obtained by computing a keyword-phones to hypothesis length

ratio, normalizing the accumulated amount of keyword phones by hypothesis length N:

$$\frac{\sum_{i=1}^{N} m(i)}{N}, m(i) = \begin{cases} 1 & \text{if } w_i \text{ is marked} \\ 0 & \text{else} \end{cases} \tag{5.3}$$

Using this keyword coverage value as additional feature for language classification, the identification accuracy significantly increased for longer test messages, whereas the performance of the short-time tests remained almost the same. The combination of both, the generic language model back-ends and the latter expansion, yielded an overall best performance for our PPR system. It was hardly possible to boost the performance on the 3s test length case, whereas the identification accuracy on longer test segments increased by up to 4% relative.
Our proposed method of keyword parsing is a very rough approximation. The missing informations about word boundaries as well as the significantly worse recognition rate compared to a word based hypothesis computation constrain the possibilities of word spotting on phone sequences. Further performance gains may be achieved by more advanced definitions of keywords, as by pronunciation dictionaries containing recognizer-generated transcriptions instead of "ideal" phone sequences. Since the phone recognizer generated hypotheses are less accurate, keywords with linguistically correct phone transcriptions are less likely to be mapped to sub-sequences of decoded message. Flawed, recognizer-dependent pronunciations may reflect the characteristics of typically generated hypotheses, thus enabling an advanced keyword coverage computation.

System	30s	20s	10s	5s	3s
PPR Baseline	91.1	90.9	89.4	87.7	86.5
PPR + Keyword LM	92.8	92.9	90.8	88.1	86.2
PPR + Keyword ratio	95.0	94.7	91.6	89.0	86.8
PPR + Keyword LM & ratio	96.9	96.6	93.8	90.5	87.0

Table 5.3: Identification accuracy (in %) of the enhanced PPR systems and each test segment length category.

5.1.2 Experiments on the PPRLM System

During system implementation, we experimented with several variations of back-end language models. One concept was to clean the corpora of all non-phone tokens, such as noise tokens, silence and filler tags. The PPRLM system equipped with language models generated from this cleaned data produced comparable results to the baseline version. This is not surprising, as the estimated hypotheses are cleaned of non-phone tokens, before they are processed by the language scoring block, i.e. the back-end language models. However, a combination of both language model types, which results in two sets of averaged language model scores, or averaged PPL respectively, seems to benefit the system's language discrimination capabilities. Regardless of the utilized score, the performance on very short test segments slightly increased by 0.4% to 0.6% absolute. The PPRLM framework using language model sentence scores achieves the best overall performance of our implemented and tested LID architectures. It yields an identification accuracy of 99.8% on 30s average test messages, and 92.4% on the average 3s category.
Further experiments, focussing on the application of expansions we tested for the PPR system, had no positive effect on the system's accuracy: Neither the keyword spotting techniques, nor the hypothesis statistics as additional features were advantageous. This is assumedly due to the worse phone recognition accuracy as a result of the pure tokenization

without language model involvement, which adversely affects the parsing for sequences keyword of pronunciations. Again, the hypothesis length and average phone duration statistics we utilized supposedly were too general to carry any additional information for language discrimination purposes.

System	30s	20s	10s	5s	3s
PPRLM Baseline (LM scoring)	99.7	99.7	98.8	96.0	92.0
PPRLM + cleaned LMs (LM scoring)	99.8	99.8	99.0	96.3	92.4
PPRLM Baseline (PPL scoring)	99.8	99.6	98.9	95.7	91.1
PPRLM + cleaned LMs (PPL scoring)	99.8	99.8	99.0	96.1	91.7

Table 5.4: Identification accuracy (in %) of the enhanced PPRLM systems and each test segment length category. Tests were done for the language model sentence scoring and perplexity measure scoring respectively.

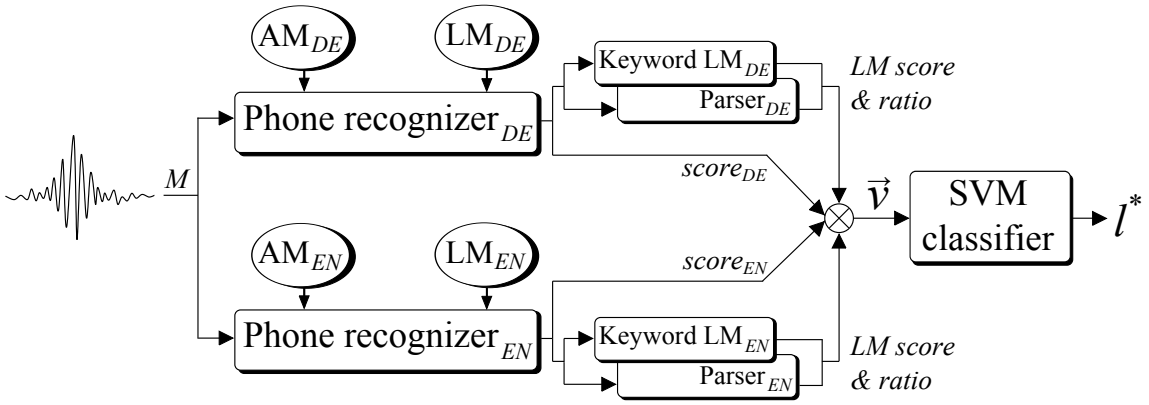

Figure 5.1: Schematic of our enhanced PPR system. Additional language cues are scores of the generic language-dependent keyword n-gram models and a keyword phones to hypothesis length ratio.

5.2 PPRLM & PPR Hybrid System

Our special interest was in the evaluation of a hybrid system, merging the PPRLM and PPR approaches. The idea was to investigate, if a combination of both architectures would result in an even more effective identification system. The system we observe uses the phone recognizers of the PPR framework as front-ends for phone recognition, followed by the PPRLM language model back-ends: The phone recognizers each generate a best hypothesis by means of acoustic and phonotactic scores, combined in the JRTk-specific recognition score. Each stream of phones is processed by the respective unmodified phone-dependent language models, which compute a language model scores or PPL scores respectively. The scores are averaged language-wise and stacked, together with the phone recognizer front-end scores, to a language cue feature vector for SVM classification.
Compared to the baseline PPR system, a significant performance boost is observable for long test data and both methods of language scoring. The LM sentence scoring back-end type yields a significantly better identification accuracy on the very short test segments, whereas the performance on the hybrid system using PPL scoring drops rapidly with shorter test durations. Despite noticeable improvements compared to the baseline and enhanced PPR systems, our final PPRLM system remains unmatched in terms of identification accuracy. Nonetheless, the hybrid approach might act as a good basis for further

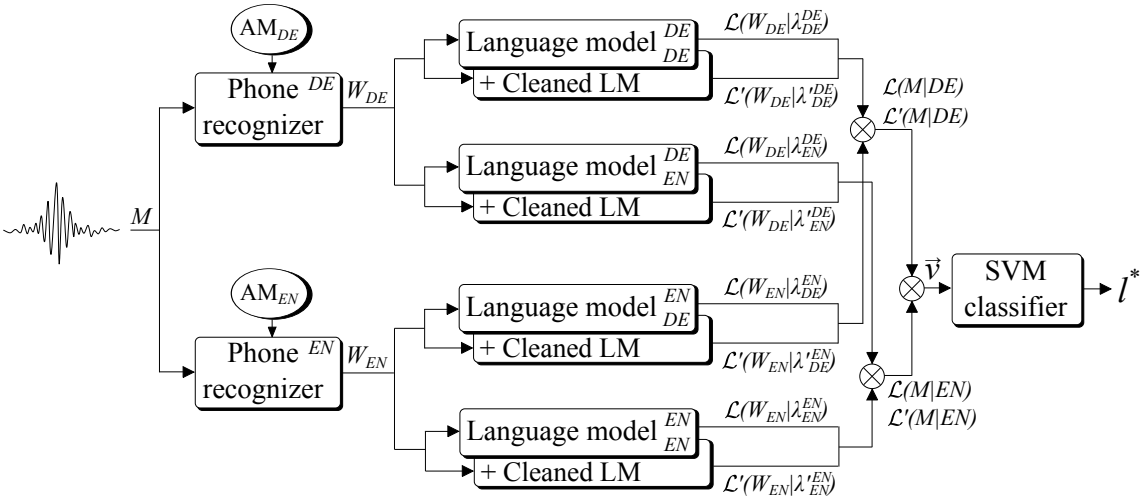

Figure 5.2: Schematic of our enhanced PPRLM system. Additional language cues are language model sentence scores of additional language models, computed upon cleaned training corpora.

research. Combining the unmodified PPR and PPRLM modules without further adaptation certainly limits the capabilities of language discrimination. The keyword spotting expansions of our PPR framework may have potential to boost a hybrid system, provided that several adaptations will raise their beneficing.

Considering the performance in terms of runtime, the hybrid system beats our best PPRLM framework, given the fact, that the complex computations of the decoders consume most of the computation time, whereas the language score computation, as well as the SVM classification run very fast. Real time factor analyses revealed, that the phone recognizers of the PPRLM system, which don't use any language models during decoding, each need about 20%-40% more computation time than the language model-incorporating recognizers used by the PPR framework. For performance measurement, we computed the real time factor

$$RTF = \frac{dur_{test}}{dur_{input}} \tag{5.4}$$

for every single phone recognizer, where dur_{test} is the accumulated test duration for all processed test utterances with an accumulated utterance duration of dur_{input}. We tested on a randomly generated test set containing test utterances with a total length of 30 minutes per language. Using the PPR front-ends gives the hybrid system a major advantage over the PPRLM architecture, enabling a potential speed-up of up to 40%. One might accept the lower accuracy of this hybrid approach for the sake of faster language identification. The PPRLM-like language model back-ends seem to compensate for the weaknesses of the pure PPR approach on short-duration testing. Hence, our proposed PPRLM & PPR hybrid outperforms even the improved PPR system with a relative increase of 5% in accuracy on the 3s test condition.

5.3 Analysis

Our experiments revealed, that the PPRLM approach outperforms the PPR architectures for all test conditions, especially on shorter test durations. Assumedly, the lower performance of the parallel phone recognition framework is induced by the shortage of the test

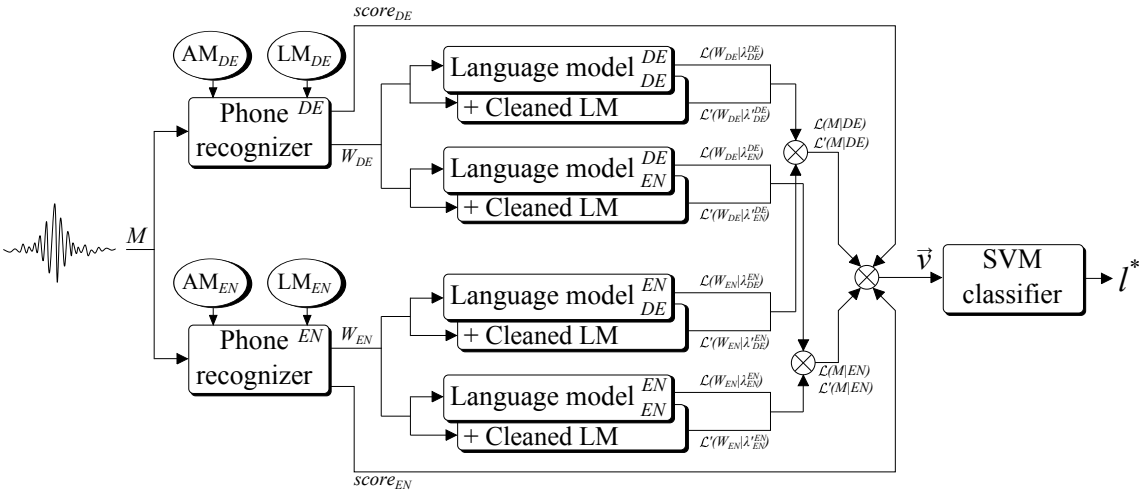

Figure 5.3: Schematic of our proposed hybrid system, utilizing the phone recognizer front-ends of the PPR approach, and the language model scoring back-ends of the PPRLM framework. The language cure vector $\boldsymbol{v}$ is generated by stacking both, the phone recognizer scores and combined and averaged language model scores.

System	30s	20s	10s	5s	3s
PPRLM & PPR Hybrid (LM scoring)	97.7	97.7	96.4	93.2	91.2
PPRLM & PPR Hybrid (PPL scoring)	98.3	98.1	95.8	91.2	86.6

Table 5.5: Identification accuracy (in %) of the PPRLM & PPR hybrid systems and each test segment length category.

messages. The scarcely available statistically relevant data seems to be insufficient for a proper identification performance. We were able to significantly boost the language identification accuracy of the baseline PPR system by applying keyword spotting techniques in form of back-end modules for the phone recognizers. A combination of a generic n-gram language model describing the language-specific keywords, and the computation of an accumulated keyword phones to hypothesis length ratio yielded the best results. However, even the capabilities of the baseline PPRLM system exceed those of the modified PPR framework. The higher phone coverage in consequence of the parallel phone tokenization followed by multiple token-dependent language models per language seems to exploit far more information from a test signal. We were able to further improve the PPRLM performance by doubling the amount of back-end language models by using additional model banks that were trained on cleaned data. The idea was to extract more channel-independent information, which may have additional language discrimination capabilities. Additionally, we tested the suitability of the perplexity (PPL) measure as substitute for the sentence score as feature for language discrimination purposes. Our experiences are, that both scoring methods perform about equally, but the commonly used sentence score proved to have slightly higher language discrimination powers on very short test segments. The utilization of PPL scores tends to result in less complex SVM models, according to the number of required support vectors. It might be, that the characteristics of these alternative scores are more suitable for the SVM training than the original log-probability sentence scores. Experiments on implementing a hybrid framework, which makes use of the PPR phone tokenizer front-ends and the PPRLM language scoring back-ends resulted in a system, which outperforms our best PPR system, but does not reach the capacities of our winning PPRLM system.

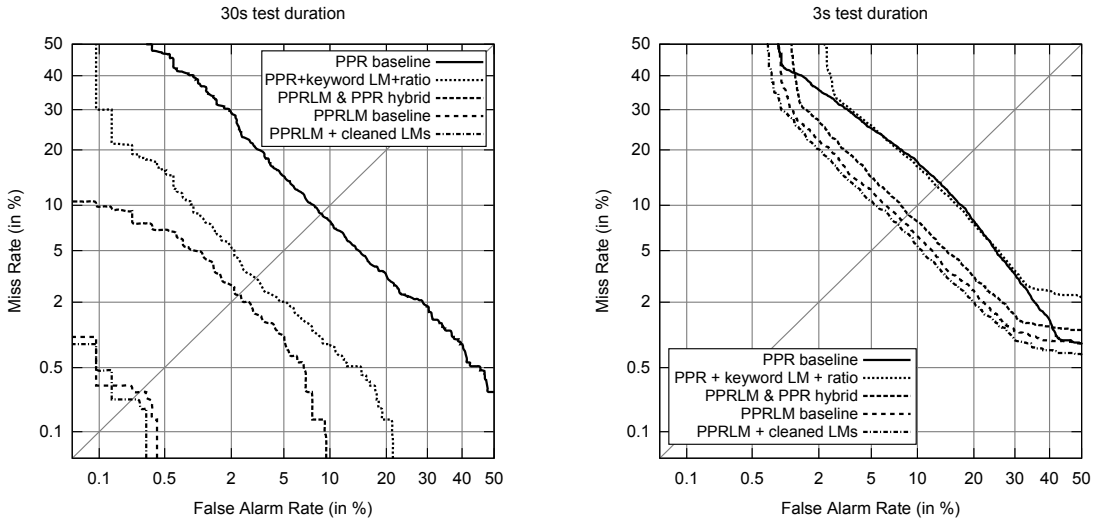

Figure 5.4: DET curves illustrating the performances of the PPR and PPRLM baseline systems and hybrid system, as well as the performance improvements of the modified systems. The PPRLM systems and PPRLM & PPR hybrid use language model scores. Depicted are the observations on the 30s test segments (left) and 3s test segments (right).

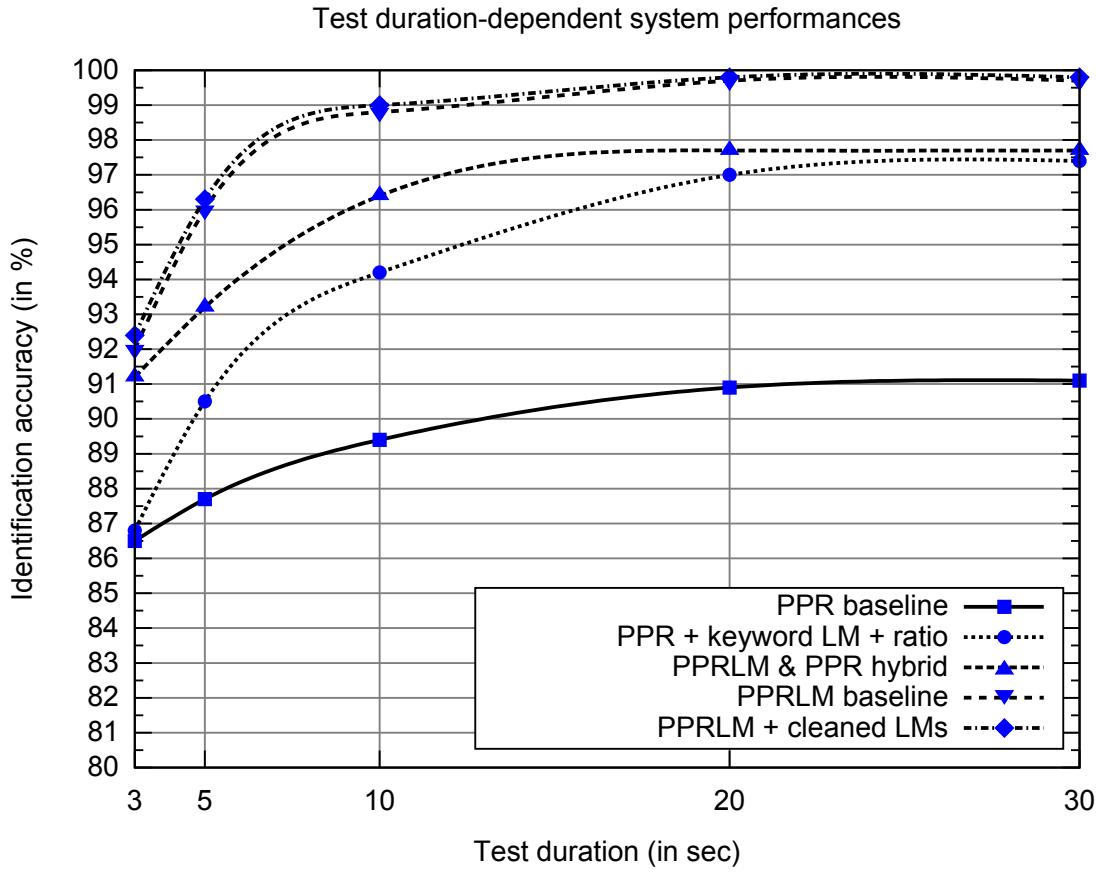

Figure 5.5: Figure showing the proposed systems' performances dependent on the test duration. The data points for the tests on segment lengths 3s, 5s, 10s, 20s and 30s are cspline interpolated.

Runtime tests, which we ran on an Intel Pentium Dual-Core E5700, 2GB RAM machine revealed, that the most complex computations take place during the message decoding process. Phone recognition runs with 1 × real-time for the PPR recognizers, and 1.5 × real-time for the PPRLM tokenizers. That qualifies both systems for real-time language identification demanding scenarios. Our hybrid system uses the superior fully fledged phone recognition front-ends. Hence, it may embody an acceptable alternative to the more accurate, but less fast PPRLM system in very time-critical applications.
The data we used for the experimental test runs is unbalanced, because we had far more German data available. In order to verify a balanced error proportion we examined the miss rates for each language, averaged over all segment length categories. Table 5.6 lists the language-specific miss rates for the three winning PPR, PPRLM and hybrid systems respectively, according to the overall performance.

System	miss_{DE}	miss_{EN}
PPR baseline	12.4	7.7
PPR + Keyword LM & ratio	7.7	10.8
PPRLM baseline (LM scoring)	3.6	3.8
PPRLM + cleaned LMs (LM scoring)	3.3	3.9
PPRLM & PPR Hybrid (LM scoring)	5.4	6.2

Table 5.6: Miss rates (in %) for each system and language pair, averaged over all segment length categories.

6. Summary

The aim of this project was to implement two standard approaches for automatic language identification using phonetic informations and phonotactic constraints. The focus during experimental test runs and evaluation was on the suitability for use in real-time identification scenarios. Thus, we aimed for reasonable performance on short-time inputs, where our minimal test durations were 3 seconds. Our baseline systems resemble the PPR and PPRLM approaches respectively. The PPRLM framework proved to be superior to the PPR system, with the latter significantly decreasing in performance on shorter input messages. We were able to boost the accuracies of both baseline systems with various expansions. Our enhanced PPR system benefits from a keyword spotting back-end based on n-gram models phone sequence parsers. The PPRLM architecture gained a slight increase in accuracy by doubling the language model back-ends with the addition of models trained on cleaned corpora. Further, an experimental PPRLM & PPR hybrid system combines modules of both approaches in a way, that it easily outperforms the PPR system, but does not reach the accuracy of the PPRLM framework. However. it is considered to be a conceivable alternative to the PPRLM system in real-time demanding applications, as it benefits from faster phone recognition.

6.1 Future Work

Our experimental results showed, that the PPRLM framework is more costly in terms of computation time, but clearly exceeds the identification accuracy of the computationally less complex PPR system. Our hybrid approach is considered a middle course with a theoretically faster identification process but at the same time a moderate loss of accuracy. In order to evaluate the performance of our proposed systems in real scenarios, further work will consist of their application as language identification front-ends for existing systems. We aim at the expansion of a system for simultaneous translation of lectures and speeches which has been developed at the KIT and embodies the first existing automatic system for simultaneous speech-to-speech translation [Füg09]. The focus of this system is the automatic translation of (technical oriented) lectures and speeches from English to Spanish [Füg09]. Hence, further tasks include the expansion of our existing LID systems to new target languages such as Spanish and French, which will lead to new evaluations regarding the identification accuracy and runtime performance in multilingual language identification scenarios.

List of Figures

List of Tables

Bibliography

[AD10] M. Adda-Decker, *Language Identification*. ISTE, 2010, pp. 279–320. [Online]. Available: http://dx.doi.org/10.1002/9780470611180.ch8

[Ass99] I. P. Association, *Handbook of the International Phonetic Association: a guide to the use of the International Phonetic Alphabet*, ser. A Regents publication. Cambridge University Press, 1999. [Online]. Available: http://books.google.com/books?id=33BSkFV_8PEC

[CL11] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[CV95] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[Füg09] C. Fügen, "A system for simultaneous translation of lectures and speeches," Ph.D. dissertation, Universität Karlsruhe (TH), jan 2009. [Online]. Available: http://digbib.ubka.uni-karlsruhe.de/volltexte/documents/1048646

[FGH+97] M. Finke, P. Geutner, H. Hild, T. Kemp, K. Ries, and M. Westphal, "The Karlsruhe-Verbmobil speech recognition engine," 1997.

[GM10] J. Gervain and J. Mehler, "Speech perception and language acquisition in the first year of life," *Annual Review of Psychology, Vol. 61, pp. 191-218, 2010*, vol. 61, pp. 191–218, 2010.

[GP03] J. J. Grieco and E. O. Pomales, "Short segment automatic language identification using a multifeature-transition matrix approach," in *ISCAS (3)*, 2003, pp. 730–733. [Online]. Available: http://dx.doi.org/10.1109/ISCAS.2003.1205123

[GW08] J. Gervain and J. F. Werker, "How infant speech perception contributes to language acquisition," *Language and Linguistics Compass*, vol. 2, no. 6, pp. 1149–1170, 2008. [Online]. Available: http://dx.doi.org/10.1111/j.1749-818X.2008.00089.x

[Haa07] H. Haarmann. C.H.Beck, 2007, ch. 1, pp. 10–16.

[Haz93] T. J. Hazen, "Automatic language identification using a segment-based approach," Master's thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1993.

[JM09] D. Jurafsky and J. H. Martin, *N-grams*, 2nd ed., ser. Prentice Hall Series in Artificial Intelligence. Prentice Hall, 2009. [Online]. Available: http://books.google.com/books?id=fZmj5UNK8AQC

[KP01] K. Kirchhoff and S. Parandekar, "Multi-stream statistical n-gram modeling with application to automatic language identification," in *INTERSPEECH*,

P. Dalsgaard, B. Lindberg, H. Benner, and Z.-H. Tan, Eds. ISCA, 2001, pp. 803–806. [Online]. Available: http://www.isca-speech.org/archive/eurospeech_2001/e01_0803.html

[LM89] M. Lorch and P. Meara, "How people listen to languages they don't know," *Language Sciences*, vol. 11, no. 4, pp. 343–353, 1989.

[LM95] ——, "Can people discriminate languages they don't know?" *Language Sciences*, vol. 17, no. 1, pp. 65–71, 1995.

[LWL+97] A. Lavie, A. Waibel, L. Levin, M. Finke, D. Gates, M. Gavalda, and P. Zhan, "JANUS-III: Speech-to-speech translation in multiple languages," in *In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'97*, 1997.

[MDK+97] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The DET curve in assessment of detection task performance," in *Proc. Eurospeech '97*, Rhodes, Greece, sep 1997, pp. 1895–1898.

[MJC94] Y. K. Muthusamy, N. Jain, and R. A. Cole, "Perceptual benchmarks for automatic language identification," in *Proc. ICASSP '94*, Adelaide, Australia, apr 1994, pp. I–333–I–336.

[Nav06] J. Navrátil, "Automatic language identification," in *Multilingual Speech Processing*. Burlington: Academic Press, 2006, pp. 233–272. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780120885015500111

[NIS09] "The 2009 NIST language recognition evaluation plan," April 2009.

[NP03] D. Nurse and G. Philippson, *The Bantu languages*, ser. Routledge language family series. Routledge, 2003. [Online]. Available: http://books.google.com/books?id=L-D0Fxw1gccC

[Rey09] D. A. Reynolds, "Universal background models," in *Encyclopedia of Biometrics*, S. Z. Li and A. K. Jain, Eds. Springer US, 2009, pp. 1349–1352. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-73003-5

[RM99] F. Ramus and J. Mehler, "Language identification with suprasegmental cues: A study based on speech resynthesis," pp. 512–521, 1999. [Online]. Available: http://cogprints.org/801/

[RQD00] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," in *Digital Signal Processing*, 2000, p. 2000.

[Sch99] H. Schütze. The MIT Press, 1999, ch. 1, pp. 39–80. [Online]. Available: http://books.google.de/books?id=YiFDxbEX3SUC

[SKN11] S. Stüker, K. Kilgour, and J. Niehues, "Quaero speech-to-text and text translation evaluation systems," in *High Performance Computing in Science and Engineering '10*, W. E. Nagel, D. B. Kröner, and M. M. Resch, Eds. Springer Berlin Heidelberg, 2011, pp. 529–542, 10.1007/978-3-642-15748-6_38. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-15748-6_38

[SMB96] V. Stockmal, D. Muljani, and Z. Bond, "Perceptual features of unknown foreign languages as revealed by multi-dimensional scaling," in *Proc. ICSLP '96*, vol. 3, Philadelphia, PA, oct 1996, pp. 1748–1751.

[SMFW01] H. Soltau, F. Metze, C. Fügen, and A. Waibel, "A one-pass decoder based on polymorphic linguistic context assignment," 2001.

[ST95] E. G. Schukat-Talamazzini, *Automatische Spracherkennung - Grundlagen, statistische Modelle und effiziente Algorithmen*, ser. Künstliche Intelligenz. Braunschweig: Vieweg, 1995.

[Sto02] A. Stolcke, "SRILM-an extensible language modeling toolkit," in *In Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002)*, 2002, pp. 901–904.

[WAC06] L. Wang, E. Ambikairajah, and E. H. C. Choi, "Multi-lingual phoneme recognition and language identification using phonotactic information," in *ICPR*, 2006, pp. IV: 245–248. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/ICPR.2006.823

[WB91] I. Witten and T. Bell, "The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression," *IEEE Transactions on Information Theory*, vol. 37, no. 4, 1991.

[Wik11] Wiktionary, "Wiktionary:frequency lists - wiktionary," 2011, [Online]. [Online]. Available: http://en.wiktionary.org/w/index.php?title=Wiktionary:Frequency_lists&oldid=13554525

[YAC08] B. Yin, E. Ambikairajah, and F. Chen, "Improvements on hierarchical language identification based on automatic language clustering," in *ICASSP*. IEEE, 2008, pp. 4241–4244. [Online]. Available: http://dx.doi.org/10.1109/ICASSP.2008.4518591

[Zis96] M. Zissman, "Comparison of four approaches to automatic language identification of telephone speech," *Speech and Audio Processing, IEEE Transactions on*, vol. 4, no. 1, p. 31, jan 1996.

[ZS94] M. A. Zissman and E. Singer, "Automatic language identification of telephone speech messages using phoneme recognition and n-gram modeling," in *Proc. ICASSP '94*, Adelaide, Australia, apr 1994, pp. I–305–I–308.