Multilingual Speech Recognition

E. Nöth and S. Harbeck and H. Niemann

Lehrstuhl für Mustererkennung (Informatik 5)
Universität Erlangen–Nürnberg, Martensstr. 3, 91058 Erlangen, Germany
email: noeth@informatik.uni-erlangen.de

Summary. We present two concepts for systems with language identification in the context of multilingual information retrieval dialogs. The first one has an explicit module for language identification. It is based on training a common codebook for all the languages and integrating over the output probabilities of language specific n–gram models trained over the codebook sequences. The system can decide for one language either after a predefined time interval or if the difference between the probabilities of the languages succeeds a certain threshold. This approach allows to recognize languages that the system can not process and give out a prerecorded message in that language. In the second approach, the trained recognizers of the languages to be recognized, the lexicons, and the language models are combined to one multilingual recognizer. Only allowing transitions between the words from one language, each hypothesized word chain contains words from just one language and language identification is an implicit by-product of the speech recognizer. First results for both language identification approaches are presented.

1 Introduction

There has been a growing interest towards language identification with the transition of speech research from laboratory systems to real life applications: consider an automatic speech understanding system for information retrieval over the telephone that is installed in Germany and that is intended to be used by the majority of the population. It will either have to be able to handle German with a wide variety of foreign accents or be able to handle German, Turkish, Greek, Italian, etc. or exclude guest workers as customers. Things get worse if the system is intended for travel information and foreign tourists are its potential customers.

In this paper we present our approach to language identification in the context of the multilingual and multifunctional speech understanding and dialog system SQEL (**S**poken **Q**ueries in **E**uropean **L**anguages). The system is being developed in the EC funded Copernicus project COP-1634. Partners are the Universities of Erlangen (Germany), Kosice (Slovak Republic), Ljubljana (Slovenia), and Pilsen (Czech Republic). The system is intended to handle questions about air flight (Slovenian system) and train connections (German, Slovak, and Czech system) in these four languages [1]. We verify our results on the SQEL corpus with two additional databases: the ATIS/EVAR [5] database with German and English sentences and the "Bundessprachenamt" corpus (BSPA) with 13 different languages.

[1] It will not be truly multifunctional in the sense that one can ask in one language questions about several applications and switch between applications during one dialog.

Basis of the system is the EVAR system, the architecture of which is based on the German SUNDIAL demonstrator (ESPRIT project P 2218) [3]. Even though major changes were made – especially in the *Linguistic Analysis* [6] and the *Dialog* module [2] – the general architecture of the SUNDIAL demonstrator was kept for the EVAR system. EVAR can handle continuously spoken German inquiries about the German IC train system over the telephone.

The rest of the paper is organized as follows: In section 2 we will explain the architecture of the national SQEL demonstrators by looking at the current EVAR system. Following this, we will motivate and introduce two different system architectures for the two versions of the integrated multilingual SQEL demonstrator. The main difference is that the first architecture (section 3.1) has an explicit language identification module, whereas in the second architecture (section 3.2), the language identification is a by–product of the speech recognition process. Following this we will explain the principle of the explicit language identification in section 3.3. In section 4 we will present results and conclude with an outlook to future work in section 5.

2 Architecture of the National SQEL Demonstrators

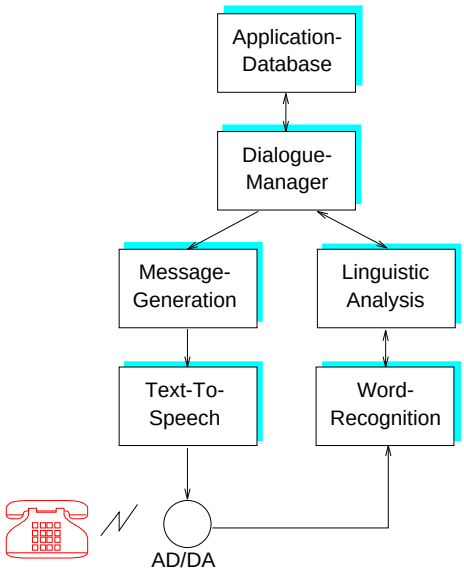

FIGURE 1. Architecture of the German SUNDIAL demonstrator.

Figure 1 shows a system overview of the German SUNDIAL demonstrator as well as of the EVAR system and the intended national SQEL demonstrators. Each of the four demonstrators will handle one language and one application. Here we

describe the EVAR system, since an improved version of it will be the German SQEL demonstrator and since it is the only SQEL demonstrator that is already fully functional. The main components of the system are:

- *Word Recognition:* The acoustic front end processor takes the speech signal and converts it to a sequence of recognized words. Ideally the recognized words are the same as what was actually spoken. Using state–of–the–art technology, the word recognizer module performs the steps signal processing, feature extraction and a search based on hidden Markov models (HMM). Signal processing techniques include channel adaptation and sampling of the speech signal. The well known mel–cepstral features as well as the first derivatives are calculated every 10 msec. Using a codebook of prototypes the first recognition step is a vector quantization. These features are used in a beam search operating on semi–continuous HMMs. Output of the module is the best fitting word chain. A description of the word recognition module can be found in [7, 11].
- *Linguistic Analysis:* The word string is interpreted and a semantic representation of it is produced. A UCG (unification categorial grammar) approach [4, 1] is used, to model the user utterances. Partial descriptions are used, which leads to higher robustness w.r.t. ill-formed word sequences. The method of delivering partial interpretations is the key to enhanced robustness of the parser. A description of the linguistic analysis module can be found in [6].
- *Dialog Manager:* This module takes the semantic representation of the user utterance and performs the interpretation within the current dialog context. It decides upon the next system utterance. Specialized modules within the dialog manager for contextual interpretation, task management, dialog control and message generation are communicating via a message passing method. A description of the dialog manager can be found in [2].
- *Application Database:* The official German InterCity train timetable database is used. Ljubljana will use the Adria Airline database, Pilsen and Kosice will use the Czech and Slovak InterCity train timetable database.
- *Message Generation* and *Text–to–Speech:* In order to have a complete dialog system this module transforms the textual representation of the system utterance into sound. We use a simple concatenation of canned speech signals (All words that the system can say are recorded and stored as individual files).

In the next section we will describe the planned adaptation steps to build an integrated demonstrator that will be able to handle dialogs in all four languages.

3 Language Identification with Different Amounts of Knowledge about the Training Data

Of course, the best language identification module is a multilingual recognizer. In speech recognition this can be implemented in the following way: starting with the speech signal, run several recognizers in parallel. Each recognizer is specialized to

one language, i.e. has an acoustic and a language model of one language. Then for each given point in time, one can identify the spoken language, based on the score (probability) for the best matching word chain in each of the recognizers. However, in this case the recognizers have to give comparable judgments. Also, if the system has to recognize *N* languages then *N* recognizers have to run in parallel, and *N-1* recognizers do work that is unnecessary for the system. Another problem with this approach is that you can only recognize these *N* languages.

Consider the situation that you want the SQEL system to be able to identify more than the four languages and react appropriately if a question is uttered in a language that can not be handled by the system. For instance, if the system identifies that an utterance was uttered in Polish, it can react with a prerecorded Polish utterance like

> The SQEL system detected a Polish utterance. Unfortunately, so far the system can only handle dialogs in Czech, German, Slovak, and Slovenian. Please ask your question again in one of these languages.

Clearly, the language identification module will not have the same quality of training data for additional languages. We might only have Polish speech samples where we know the language, but not what was said. Also, the samples might be from a very different domain, and the other necessary resources (pronunciation lexicon, stochastic language models) might not be available.

Our strategy for integrating the national demonstrators into one system is twofold:

- Build a system with explicit language identification. The only label of the training data for the language identification is the spoken language. The topic or the spoken words of the training utterance will not be known. We will describe the architecture of this system in section 3.1.
- Develop a multilingual recognizer for the *N* languages. In this case the same amount of labeled training data and resources (pronunciation lexicon, stochastic language models) has to be available for the languages to be identified as for the languages to be recognized. The language identification is done implicitly during the decoding of the utterance. We will describe the architecture of this system in section 3.2.

3.1 A System with Explicit Language Identification

Figure 2 shows a system overview of the intended final SQEL demonstrator with explicit language identification. As can be seen, the major changes affect the word recognition module and the information flow between the modules. Since we plan to use as many software modules as possible from the EVAR system, many of the internal changes can be implemented via switches for language specific resource files. To do this, the modules have to have a control channel in addition to the existing data channel. The control channel will be used to pass messages like identity of the language and current application. The four–way arrows in Figure 2 indicate switches, the double arrows indicate data flow and the single arrows indicate control flow. The

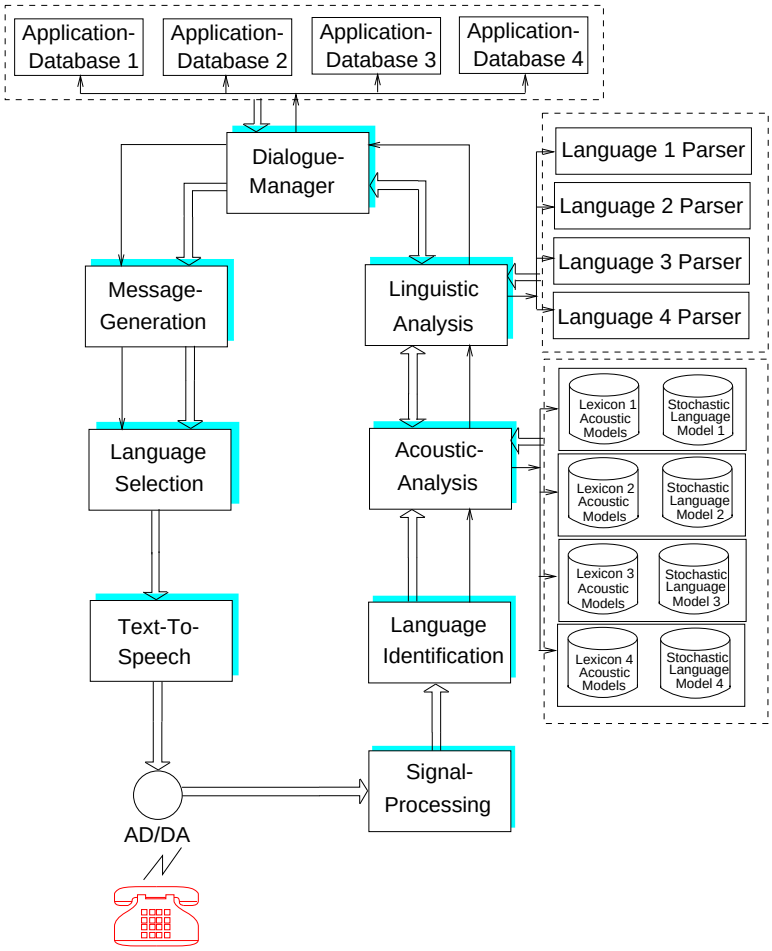

FIGURE 2. Architecture of the SQEL demonstrator with an explicit language identification module.

signal processing can be done independent of the language. The next steps — vector quantization and HMM search — need language dependent data. What is needed are language dependent codebooks, lexicons and stochastic language models. If the module has information about what language was uttered, it can simply switch to the resource files of the right language. Therefore a *language identification* module has to be added to the system that has to identify the language and pass a message to the remaining modules. The module will be activated at the beginning of the dialog. To save computation time, we use the same mel–cepstral features as the recognition module. After a certain time interval a classification step between the N languages is performed. The application requires a decision after only a few seconds,

because users tend to make short queries [2]. An overview of algorithms for language identification is given in [9, 13]).

During the analysis of further user utterances the language identification module simply passes on the extracted feature vectors and causes no delay.

Clearly, there is a tradeoff between recognition accuracy and delay time for the task of language identification: The longer the utterance, for which the sequence of feature vectors is computed, the more language specific sounds have been uttered by the caller and the better the automatic language identification will be. On the other hand, the recognition has to wait for the language identification decision, before it can start. As mentioned above, it is not clear yet, how long the utterance has to be for languages as close as Czech and Slovak, in order to be able to classify the language at an acceptable rate. This leads us to an alternative approach presented in the next section.

3.2 A System with Implicit Language Identification

We want to use all knowledge sources that are available as early as possible, i.e. apply n speech recognizers for the language identification process. To reduce the computational load mentioned above, we build a recognizer that contains all words from all languages in its dictionary. By using a stochastic bigram language model that only allows transitions between words within one language, each hypothesized word chain will only contain words of one language.

The basis for our multilingual speech recognition system are monolingual speech recognizers. We use semi-continuous HMMs for acoustic and bigrams for linguistic modeling. The monolingual recognizers are trained in the ISADORA [10] environment which uses polyphones with maximum context as subword units. The construction of the multilingual speech recognizer is as follows:

1. Increase the number of codebook density functions to reflect the language dependent codebooks. For example when having two different languages with a codebook of 256 density functions per language, then the multilingual recognizer will have 512 density functions.
2. Add special weight coefficients to the HMM output density functions to reflect the increased number of available density functions. The new weight coefficients are set zero, so that every density function belonging to different languages has no influence on the output probability of the HMM.
3. Construct a special bigram model which consists of the monolingual bigrams and does not allow any transitions between the languages as shown in equation 1.

$$P(\text{word}_{language_i}|\text{word}_{language_j}) = 0 \quad \text{for } i \neq j. \tag{1}$$

Figure 3 shows the alternative system architecture. One might argue that this approach will slow down the recognition, just like running *N* smaller recognizers in parallel, since we quadruple the lexicon. At the beginning of the recognition process every word of the multilingual vocabulary is possible, so that there are a lot of different search paths. After a few seconds the most probable paths will be in the

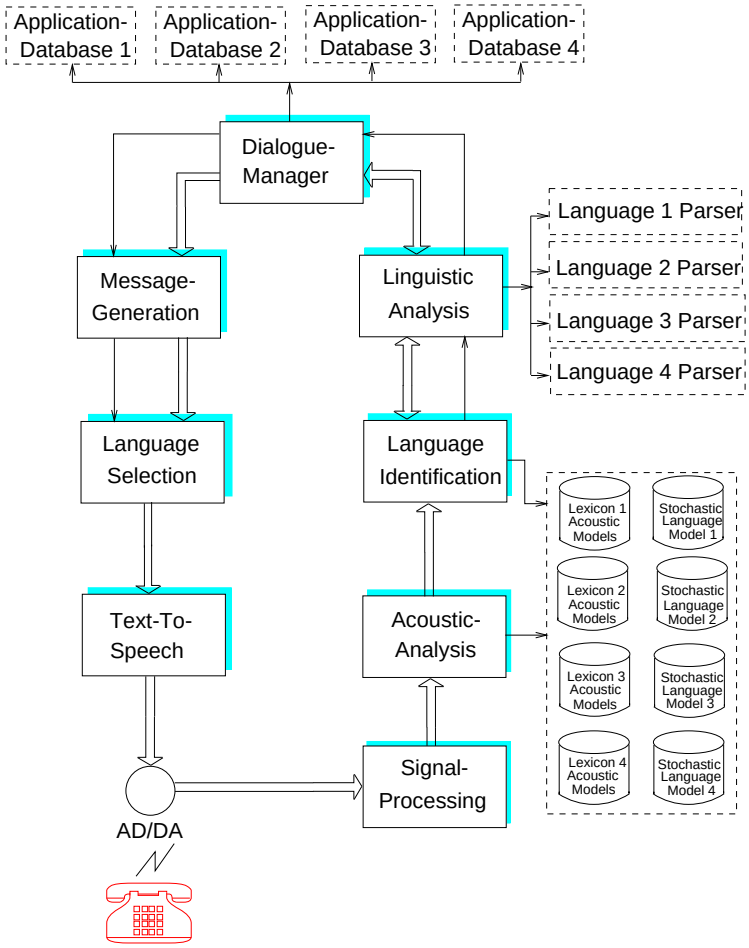

FIGURE 3. Architecture of the SQEL demonstrator with implicit language identification.

correct language. The acoustic models of the other languages should result in paths with lower scores. The beam search algorithm [8] is used to restrict the search space to paths through the word hypotheses graph which contain more reliable hypotheses. Experiments showed that this suboptimal search strategy has no bad effects on the word recognition rate. So using the beam search strategy in forward decoding only paths of the correct language should be expanded. After a few words it should be as fast as the monolingual speech recognition system. In Figure 4 the number of states inside the beam for the multilingual and the monolingual speech recognizers are compared for one English sentence. At the beginning of the sentence all available languages are possible. Therefore, the number of states is significantly higher than in the monolingual case. After a short time (less than 2 seconds) all states of the

wrong language are pruned and the number of states inside the beam is the same as the one for the monolingual recognizer. Therefore the increase in computational load is neglectable while for running N recognizers in parallel it increases by more than N (see section 4).

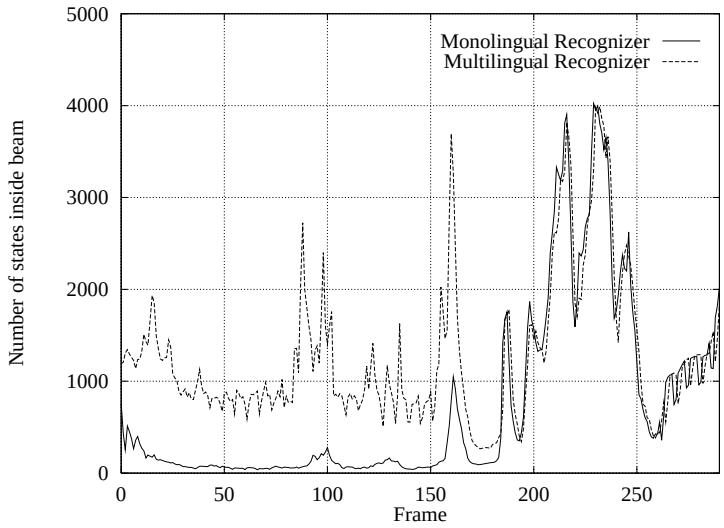

FIGURE 4. The number of states inside the English monolingual and the German/English multilingual speech recognizer evaluated on an English sentence. At the beginning of the sentence the multilingual recognizer has much more states inside the beam because all languages are still possible. After 200 frames (2 seconds) all states of German models are pruned and the number of states of both recognizers are the same.

In the next section we will describe the language identification module that will be used in the first system architecture. The implementation of the implicit language identification for the second system is straight forward and we will not further elaborate on it.

3.3 Language Identification Based on Cepstral Feature Vectors

We want to build a module that only knows the identity of the training utterances, because we want to train additional languages. In order to be as efficient as possible, we want to use as many processing steps of our speech recognition system as possible. The following steps are performed:

- Extract the same mel-cepstral features and derivatives as for the recognition task. Thus after the identification no new feature extraction is necessary.
- Take an appropriate subset of the features. The lower cepstral coefficients are more sound specific, i.e. language specific, whereas the higher coefficients are

more speaker specific. In preliminary experiments [12] good results were achieved with using the first six mel–cepstral coefficients from three consecutive frames resulting in a feature vector of length 18.
- Train a vector quantizer with the training data from all the languages together. Output of the vector quantizer is a sequence of indices, i.e. we use a hard vector quantizer.
- Train N n–gram language models over the symbol sequences for the N languages.
- To identify the language, calculate the sequence of vector quantizer symbols and calculate the N n–gram probabilities in parallel. For each language sum up over the sequence of the negative logarithms of the n–gram probabilities. At any time the algorithm can then decide for the most likely language.

Note that for large values of N a beam search can be used, i.e. after a certain interval, languages that are below a certain threshold, are discarded. Also, the module can decide for the language with the highest probability either after a fixed time interval or if the difference between the best and the second best alternative exceeds a certain threshold.

4 Results

In this chapter we want to present some preliminary results of explicit and implicit language identification experiments based on the SQEL , the ATISand the BSPA corpus.

Results for Explicit Language Identification

For the explicit language identification we tried to recognize the three Slavic SQEL languages Czech, Slovenian and Slovak. We trained the quantizer and the three n–gram language models with 4 hours of speech (1.7 hours from 42 Slovenian, 1.5 hours from 30 Slovak, and 1 hour from 23 Czech speakers).

Line “w/o LDA” in Table 1 shows recognition rates of the explicit language identification module for 17 minutes from 9 independent test speakers (4 Slovenian, 3 Slovak, and 2 Czech speakers). We achieve even better results when using a transformation of the feature space with the *Linear Discriminant Analysis* (Line “with LDA” in Table 1)

Considering the small amount of training data, the similarity of these three languages and the time to decide, these results are very encouraging. However, it should be kept in mind, that so far we used high quality speech input and that the speech material is read speech from a restricted domain. Nevertheless, at least the restricted domain is realistic for an application in a human machine dialog system.

To prove our results even on noisy and inhomogeneous data we evaluated our algorithm on a different corpus, which was collected by the Federal Institute for Language Engineering (Bundessprachenamt) in Germany. This corpus was collected via television and radio and contains 13 different languages (German, English,

	rec. rate for Czech	rec. rate for Slovenian	rec. rate for Slovak
w/o LDA	91.47 %	98.67 %	93.65 %
with LDA	96.76 %	97.95 %	98.67 %

TABLE 1. Recognition rate for explicit language identification between three languages with and without LDA feature transformation. Forced decision after 2 seconds (or at the end of the utterance, if it is shorter than 2 seconds).

Arabian, Chinese, Italian, Dutch, Polish, Portuguese, Rumanian, Russian, Swedish, Spanish and Hungarian). Contrary to the SQEL database the recording conditions are very variable, the domain is not restricted and spontaneous and read speech are mixed. To keep the amount of training data constant for each language we use half an hour of speech per language and the rest of data for testing (between 10 minutes and two hours, 30 minutes in the average). Therefore the recognition rates given in Table 2 are class wise averaged. The perplexity of the BSPA corpus was much higher than in the restricted domain of the SQEL corpus, leading to a higher number of different observed n–grams. Nevertheless Table 2 shows that the approach works even under worse conditions, for example a class wise averaged recognition rate of 76 percent is achieved when 60 seconds segments are classified. Nevertheless the results show that when the number of languages or the complexity of the task which is handled by the system increases the necessary time to identify the language is not acceptable for the overall dialogue system. This leads us to the approach with implicit identification which is evaluated in the next section.

rec. rate for 2s	rec. rate for 10s	rec. rate for 30s	rec. rate for 60s
34.8 %	55.7 %	68.5 %	76.0 %

TABLE 2. Class wise recognition rates for explicit language identification between 13 language on the BSPA corpus for different segment lengths.

Results for Implicit Language Identification

We tested our multilingual speech recognizer approach on two different databases. The first database is the Slovak and Slovenian corpus of the SQEL project, the second contains spontaneous corpora for German (EVAR) and English (ATIS) (see also Table 3).

In our first experiment we took the baseline system as described in section 3.2 and evaluated it on the SQEL database (see Table 4, row multilingual). The word accuracy of the Slovenian sentences is the same as in the monolingual recognizer, but the accuracy for Slovak sentences decreased by 60 percent. The problem is that

Language	Amount of training data	# of speakers (calls)	Amount of test data	# of speakers (calls)	Level of spontaneity
Slovak	4.5 h	30	40 min	4	read
Slovenian	4.5 h	42	40 min	6	read
German	7 h	804	1 h	234	spontaneous
English	7 h	46	2 h	30	spontaneous

TABLE 3. Description of training and test sets used for the multilingual speech recognizer.

the beam search often canceled all states of the correct language after a short time in almost all Slovak sentences. Once there are no Slovak states inside the beam, the recognizer cannot return to the Slovak language. When using two different beams inside the forward decoding, one very big beam for the first part of an utterance and one normal one for the rest of the utterance, we increased the recognition rate of Slovak with the side effect of higher computation time.

Row “multilingual with multilingual silence” in Table 4 and Table 5 show the effect of using a multilingual silence category. Instead of using different silence models for each language all silence models for all languages are in one common category. This method allows transitions between the languages by using a silence model during decoding. The word accuracy using the multilingual word recognizer was almost as good as using the correct monolingual recognizers in both databases. Additionally the computation time was almost as good as using the correct monolingual recognizer, whereas the time of a recognizer evaluated on an out-of-language speech signal increases drastically: if one speaks a sentence in a foreign language into an automatic speech recognition system, the recognition time generally increases significantly, because nothing matches well and thus the dynamically adapted beam width [7, p. 120] goes up. Table 5 shows the computation time for the monolingual and the multilingual recognizers.

Recognition rates (word accuracy)		
Monolingual Slovenian	91 %	
Monolingual Slovak		86 %
Multilingual	91 %	29 %
Multilingual with multilingual silence	90 %	87 %

TABLE 4. Recognition rates for the multilingual word recognizer and the monolingual recognizers in the languages Slovak and Slovenian.

Table 6 shows the same results hold when running a bilingual German/English recognizer. Again the word accuracy stayed practically the same.

Computation time with multilingual silence		
Recognizer	Slovenian	Slovak
Monolingual Slovenian	20 min	1 h
Monolingual Slovak	1 h	20 min
Multilingual	20 min	20 min

TABLE 5. Computation time for the multilingual word recognizer and the monolingual recognizers in the languages Slovak and Slovenian.

Recognizer	**WA on English**	**WA on German**
Monolingual	63 %	71 %
Multilingual	64 %	71 %

TABLE 6. Recognition rates for the multilingual and the monolingual recognizers on the ATIS/EVAR task.

5 Conclusions and Future Work

We presented two concepts for systems with language identification in the context of multilingual information retrieval dialogs. The first architecture is a straightforward integration of an explicit language identification module. It has the advantage of being able to recognize languages that can not be processed by the system and allows an appropriate reaction. It has the disadvantage of delaying the recognition process until the spoken language can be identified with a high accuracy. The alternative approach is to combine the monolingual recognizers to one recognizer. By forcing word transitions to stay within one language, the system identifies the language and decodes the utterance simultaneously. Since the beam search eliminates partial hypotheses with bad scores, the size of the search space approaches that of the monolingual recognizers. Thus, the delay caused by increased vocabulary size is small. The approach utilizes the available speech data more efficiently than the explicit language identification, but can not identify additional languages.

For the explicit identification preliminary experiments with the three Slavic SQEL languages were presented that showed that the language can be identified with high accuracy after only two seconds.

For the implicit identification we presented first results with a bilingual recognizer for Slovenian and Slovak and for English and German, indicating that the combined system can achieve the same recognition rates on both languages as the two monolingual recognizers. The time behavior also stayed the same as for the monolingual recognizers whereas the time behavior of the monolingual recognizers on the wrong language showed that our approach is superior to running N recognizers in parallel for language identification purposes.

In the future we plan to extend the number of different languages inside the multilingual recognizer. Because with an increasing number of languages the number

of output probabilities is growing, we want to examine an approach of sharing the same codebook or the same acoustic models for subword modeling.

6 Acknowledgment

This work was partly funded by the European Community in the framework of the SQEL–Project (Spoken Queries in European Languages), Copernicus Project No. 1634. The responsibility for the contents lies with the authors.

7 REFERENCES

[1] F. Andry, N. Fraser, S. McGlashan, S. Thornton, and N. Youd. Making DATR Work for Speech: Lexicon Compilation in SUNDIAL. *Computational Linguistics*, 18(3):245–267, Sept. 1992.

[2] W. Eckert. *Gesprochener Mensch–Maschine–Dialog*. Berichte aus der Informatik. Shaker Verlag, Aachen, 1996.

[3] W. Eckert, T. Kuhn, H. Niemann, S. Rieck, A. Scheuer, and E. G. Schukat-Talamazzini. A Spoken Dialogue System for German Intercity Train Timetable Inquiries. In *Proc. European Conf. on Speech Communication and Technology*, pages 1871–1874, Berlin, 1993.

[4] R. Evans and G. G. (eds.). The DATR Papers: February 1990. Technical report, Cognitive Science Research Paper CSRP 139, University of Sussex, Brighton, 1990.

[5] C. H. J. Godfrey and G. Doddington. The ATIS Spoken Language Systems Pilot Corpus. In *Speech and Natural Language Workshop*, pages 96–101. Morgan Kaufmann, Hidden Valley, Pennsylvania, 1990.

[6] G. Hanrieder. *Inkrementelles Parsing gesprochener Sprache mit einer linksassoziativen Unifikationsgrammatik*. PhD thesis, Universität Erlangen–Nürnberg, 1996.

[7] T. Kuhn. *Die Erkennungsphase in einem Dialogsystem*, volume 80 of *Dissertationen zur Künstlichen Intelligenz*. Infix, St. Augustin, 1995.

[8] B. Lowerre and D. Reddy. The Harpy Speech Understanding System. In W. Lea, editor, *Trends in Speech Recognition*, pages 340–360. Prentice–Hall Inc., Englewood Cliffs, New Jersey, 1980.

[9] Y. K. Muthusamy, E. Barnard, and R. A. Cole. Reviewing automatic language identification. *IEEE SIGNAL PROCESSING MAGAZINE*, pages 33 – 41, Oktober 1994.

[10] E. Schukat-Talamazzini, T. Kuhn, and H. Niemann. Speech Recognition for Spoken Dialogue Systems. In H. Niemann, R. De Mori, and G. Hanrieder, editors, *Progress and Prospects of Speech Research and Technology: Proc. of the CRIM / FORWISS Workshop*, PAI 1, pages 110–120, Sankt Augustin, 1994. Infix.

[11] E. G. Schukat-Talamazzini. *Automatische Spracherkennung – Grundlagen, statistische Modelle und effiziente Algorithmen*. Vieweg, Braunschweig, 1995.

[12] V. Warnke. Landessprachenklassifikation. Studienarbeit, Lehrstuhl für Mustererkennung (Informatik 5), Universität Erlangen–Nürnberg, 1995.

[13] M. Zissman. Comparison of four approaches to automatic language identification of telephone speech. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 4:31–44, 1996.