INTEGRATION OF A KALDI SPEECH RECOGNIZER INTO A SPEECH DIALOG SYSTEM FOR AUTOMOTIVE INFOTAINMENT APPLICATIONS

Thomas Ranzenberger[1,2], *Christian Hacker*[1], *Florian Gallwitz*[2]

[1]*Elektrobit Automotive GmbH,* [2]*Technische Hochschule Nürnberg Georg Simon Ohm*
thomas.ranzenberger@elektrobit.com

Abstract: In this paper we present an evaluation of the Kaldi speech recognizer in an automotive context. We integrate Kaldi into an existing software tool which is used to specify human-machine interfaces including speech dialogs for automotive and non-automotive domains. This enables linguists and other researchers to use their own Kaldi models for user studies or experiments on voice enabled interfaces. We train our own Kaldi models using a freely available corpus based on audiobooks. Further, we propose an algorithm to map utterances returned by the Kaldi recognizer onto intents of the speech dialog system. We evaluate the provided algorithm with our trained Kaldi model in the extended software tool. The used Kaldi model is based on time delayed neuronal networks and has a word error rate of 5.9% and a sentence error rate of 52.5% on the test data of the corpus. 22 participants spoke 50 random sentences of a self created corpus of example sentences. The words of the collected corpus are a subset of the words which are used for the language model of the Kaldi recognizer. The applied method is able to reduce the sentence error rate from 34% to 3% on this corpus. The Kaldi speech recognizer is suitable for automotive command and control scenarios. The recognized intents are detected robustly with the used algorithm and proposed modeling techniques.

1 Introduction

Speech user interfaces are nowadays an essential part of a car. The user interface in the car is more and more influenced by the current trend for personal assistants in smart homes and smartphones. The complex technology and growing functions of today's cars require the development of a well designed human machine interface (HMI) combined with an intuitive speech dialog system. Most of the speech recognizer in the field of automotive are proprietary. Therefore we would like to evaluate an open source speech recognizer in an automotive context. For our investigations we used the open source toolkit *Kaldi* [1]. It enables the research and development of speech recognition technologies and applications. In the following sections we show related work, use cases, and methods how a Kaldi speech recognizer could be integrated into an existing dialog system. After it we define a typical automotive application scenario and derive needs for our approach before presenting our way to process sentences for a command and control system. Finally we describe an experiment to evaluate our method. The aim of the method is to lower the sentence error rate in the described scenario.

2 Related work

In this publication we use an algorithm to calculate the distance between the recognized utterance and example phrases. The publication of Ye and Young [2] uses a dynamic time warping

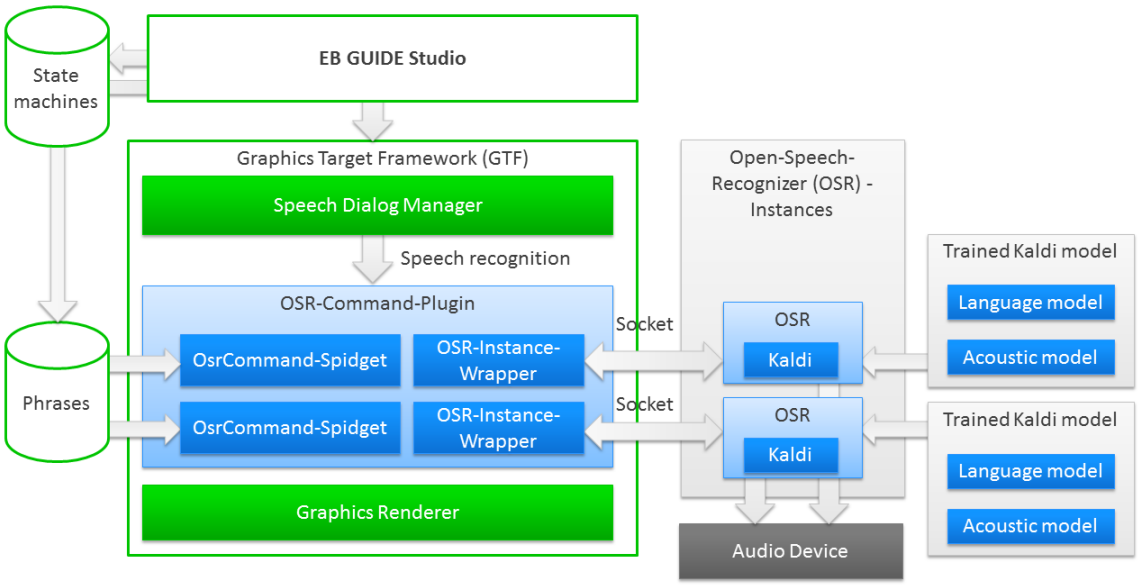

Figure 1 – Integration of the Open-Speech-Recognizer (OSR) based on the Kaldi toolkit into EB GUIDE which consists of the modeling tool EB GUIDE Studio and the runtime framework GTF. The OSR-Command plugin extends the Speech Dialog Manager and provides Spidgets to the modeling tool.

(DTW) algorithm as distance for a k-means clustering method for semantic decoding. The DTW calculates a distance between two strings. The distance is also known as Levenshtein distance. It is defined as the minimum number of insertions, deletions or substitutions required to change one word or tree into the other [3]. Alternative approaches for semantic analysis used in the field of spoken language understanding are published by Tur and De Mori [4]. Another paper with the focus on spoken language understanding in embedded systems was published by Weilhammer et al. [5].

3 Integration of Kaldi into a dialog modeling tool

The speech dialog system for our research is part of the HMI modeling tool *EB GUIDE* [6]. It is widely-used in the automotive industry, in particular to create infotainment systems. The dialog flow is modeled using a state machine. Each state represents a *talk* to specify a dialog step. The tool provides different command and prompt *spidgets* (speech gadgets) to handle speech recognition and speech output (e.g. TTS) within a talk. The specified dialog flow and user interface can be simulated directly of exported for different target platforms [6].

The Kaldi toolkit [1] is integrated via an own speech recognizer application Open-Speech-Recognizer (OSR) which uses the Kaldi libraries. The OSR is able to load trained Kaldi models, streams the audio signal of a microphone, and performs speech to text decoding. It supports classical GMM-HMM approaches and hybrid approaches like TDNN-HMM by using the built-in implementations to decode Kaldi models [1]. Furthermore, OSR provides a socket interface to enable control of the recognizer by an external application like our speech dialog system. Figure 1 shows how multiple OSR instances are connected via a socket interface to the modeling software. The OSR is able to run on Windows or Linux operating systems. Embedded devices based on QNX or Linux are widely used in today's car information systems.

The modeling tool and the runtime executable is extended with our OSR-Command-Plugin to integrate the recognition module into the dialog flow and to control the OSR application. The plugin enables the usage of multiple instances of OSR applications which allows to use multiple Kaldi models within the dialog system. The instances are managed by the OSR-Instance-Wrappers. thus, an acoustic and language model for digit recognition and models

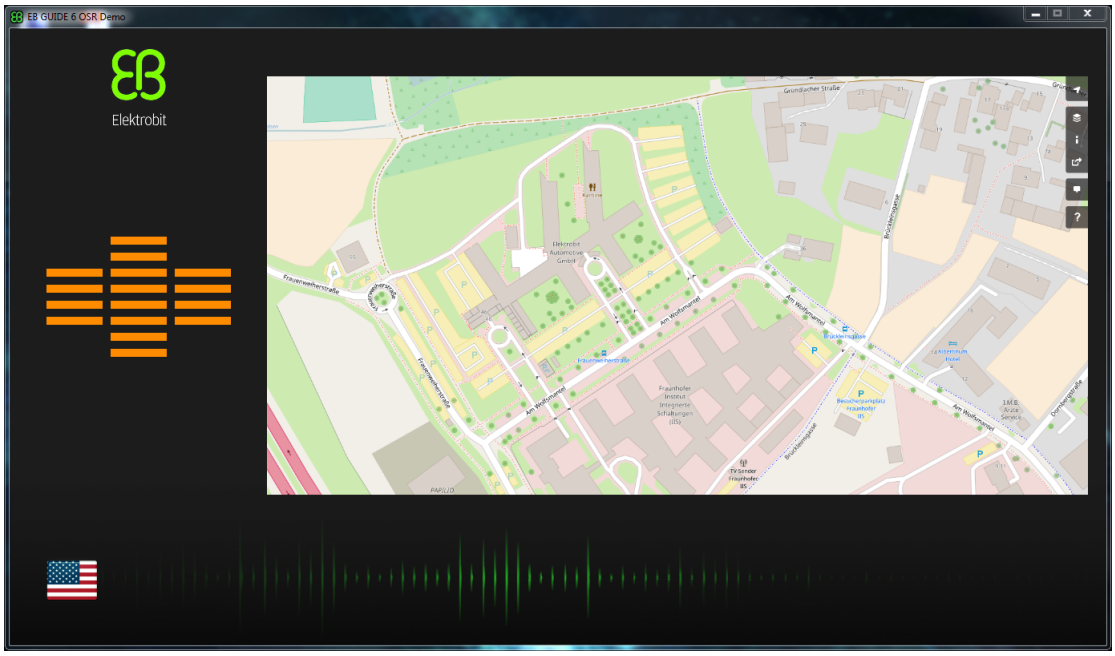

Figure 2 – Simulation of an EB GUIDE model with navigation screen displayed after recognizing "start navigation"

for large vocabulary recognition can be preloaded at startup of the EB GUIDE runtime and used in different dialog states. The ability to load, unload, and preload Kaldi models helps to handle limited resources of the target system. The OSR-Command-Plugin further provides a new command spidget *OsrCommand* to the modeling tool in order to specify an intent and to configure Kaldi parameters and the used Kaldi model in a specific talk. The spidget is able to access the EB GUIDE datapool which contains all global variables like the phrases we need in our approach [6].

4 Automotive scenario

In the following we describe an typical automotive scenario which we then model with the EB GUIDE tool. A state of the art infotainment system consists of several applications like radio, browser, navigation, and phone. In addition the system needs to handle dynamic data like stationlists or large set of several thousand phone contacts. The system reacts on a wake-up word to initiate a speech dialog, which is in our case *hello eva* (Elektrobit Virtual Assistant). Table 1 shows command and control example phrases and the resulting actions of the infotainment system. The first phrase starts the route guidance of the navigation application and displays the map like it is shown in Figure 2.

Table 1 – Possible example phrases and actions in the automotive scenario

Phrase	Intent	Action	Possible follow-up phrase
start navigation	navigation	show map	
start web search	web_search	open browser app	
dial a number	phone_call	open phone app	09110815
set radio station	station_change	open radio app	Star FM

To control the infotainment system it is necessary to map the recognized utterances to intents. This mapping requires additional methods in the speech dialog system to process the results of the Kaldi recognizer. The approach we used is described in the following section.

5 Approach

To implement the described scenario, the speech dialog system needs to be extended. As a first step, our OSR application needs to be able to load Kaldi models and to return the recognized utterances to the speech dialog system. We train different Kaldi models based on the librispeech corpus. It is a large corpus (1000 hours) of English read speech derived from audiobooks [7]. We use the existing recipe provided by Povey et al. [8] using time delayed neuronal networks to create a Kaldi model. We also try to train additional neuronal network based Kaldi models. The language model contains up to 200000 words.

As a second step, the recognized utterance of the OSR needs to be mapped onto the right action. Therefor we need to classify the intent of the utterance. We provide several methods to compare the utterance with a corpus of domain specific example phrases tagged with their intent. Initially, a simple lookup method for specific words or phrases was chosen. In addition a simple lookup method for exact phrases was implemented. But the simple approaches were not enough to fulfill the needs for the automotive scenario: The intent recognition on sentences with the previous methods didn't worked well. To reduce the sentence error rate we implemented a dynamic time warping (DTW) algorithm. It compares the recognized utterances with the provided example phrases. It calculates the distance between each utterance and each example phrase and decides for the intent belonging to the phrase with lowest distance.

Table 2 shows a subset of phrases which are used for the intent "navigation". Table 3 shows the recognized utterances returned by the OSR. The implemented algorithm calculates a distance of zero for the third phrase from Table 2 and the fifth utterance of the n-best list in Table 3 and triggers the right action to show the navigation map.

Table 2 – Example phrases for an intent "navigation"

Phrase	**Action**
start navigation please	show map
show navigation	show map
start navigation	show map
navigate to destination	show map

Table 3 – Utterances returned from OSR after recognition (n-best list) and distance to the example phrase "start navigation"

Utterance	**Distance**
STUART NAVIGATION	1
STARTED NAVIGATION	2
STARK NAVIGATION	1
STORED NAVIGATION	3
START NAVIGATION	0
STARRED NAVIGATION	2
STORK NAVIGATION	2

In the modelling tool we use multiple instances of the OsrCommand spidget to configure the recognition. Each command spidget represents an intent and can be configured with OSR recognizer settings, with example phrases, and with an action to react in the model. For each spidget the local minimum distance of the utterances to a list of example phrases is calculated. The spidget with the lowest global distance wins and its action is triggered. Figure 3 shows a

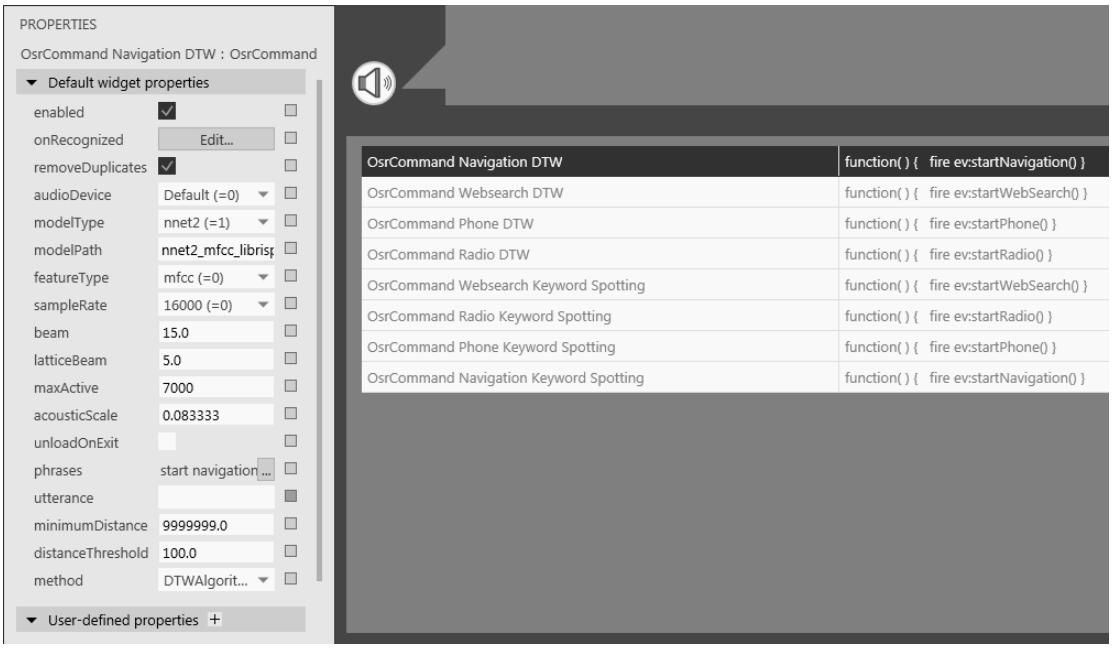

Figure 3 – Talk with multiple OsrCommand-Spidgets to recognize intents (right). The action to the selected intent "navigation" is a script firing an event processed in the graphical part of the model. On the left side you see properties of the selected spidget, i.e. "phrases" with a list of example phrases and "utterance" with a link to the global datapool, where the recognized utterance is written into during runtime.

screnshot of our modeling tool that explains how parallel active intents are configured as a list of spidgets clustered in a talk state. The described usage of the DTW algorithm with multiple spidgets is evaluated in an experiment described in the next section.

6 Experiment

To evaluate our previously described approach we created a new EB GUIDE evaluation model which uses the OSR with a trained Kaldi model. The used Kaldi model is trained on the librispeech corpus [7] with TDNN. We also tried to train a bidirectional long-short term memory (BLSTM) [9] and a combination of TDNN and BLSTM. Table 4 shows the word error rates of the trained models. The BLSTM models are consuming more memory than the TDNN model. The initialization time of the BLSTM models was higher than the TDNN model. We decided to use the TDNN model for our automotive scenario because memory and performance are important factors on embedded systems. Additionally, the lowest word error rate was achieved with TDNN.

For the following experiment we use the TDNN based model and the `tgsmall` language model for the recognition. We use a 40 dimensional vector of filterbank features stored as Mel-Frequency Cepstrum Coefficients (MFCC) as input and a sample rate of 16 kHz. The beam parameter is set to 15 and the lattice beam to 8. We use a maximum of 7000 active states for the decoding. The acoustic scale parameter was set to one. The `tgsmall` language model is based on tri-grams (`tg` in the data column of Table 4) and has still an acceptable word error rate. It is pruned and consumes less memory at runtime than the larger language models like the four-gram (`fg` in the data column of Table 4) based model `fglarge` (`large` language model) or the tri-gram based model `tglarge`. The `small` language model was chosen because we assume that in our automotive scenario we have limited resources on the target platform. Table 4, right column, shows the sentence error rate of the TDNN model on the librispeech corpus. In our experiment we evaluate the sentence error rate of the used Kaldi model on the automotive domain and evaluate how the usage of the DTW algorithm affects the sentence error rate.

Table 4 – Word error rates (WER) and sentence error rates (SER) of trained neuronal networks for the librispeech corpus. Data indicates the used audio data (clean vs. other) of the librispeech corpus and the language model (tg: tri-gram, fg: four-gram, small/large: size after pruning).

Data	**WER**			**SER**
	TDNN	**BLSTM**	**TDNN-BLSTM**	**TDNN**
test_clean_fglarge	4.15%	7.46%	6.62%	41.15%
test_clean_tglarge	4.42%	7.73%	6.97%	42.79%
test_clean_tgmed	5.29%	9.50%	8.28%	48.36%
test_clean_tgsmall	5.91%	10.19%	8.98%	52.52%
test_other_fglarge	10.56%	18.80%	17.50%	64.78%
test_other_tglarge	11.13%	19.53%	18.15%	66.45%
test_other_tgmed	13.35%	22.32%	20.46%	72.71%
test_other_tgsmall	14.80%	23.85%	21.79%	76.08%

To evaluate the DTW algorithm we use an own automotive corpus *EB-Car*. The corpus was collected for a car infotainment scenario. It contains 459 words and 973 sentences in English. Sentences contain commands like *start navigation* as shown in Table 1. The words of *EB-Car* are a subset of the words which are used for the librispeech corpus based Kaldi model. The corpus was collected by asking 27 people to write down possible questions to an infotainment system.

Each participant of our evaluation was asked to speak 50 random sentences out of the *EB-Car* corpus. The participants are using a headset. The DTW algorithm compares the recognized utterance with the complete set of phrases from the *EB-Car* corpus. We count the sentences correctly identified by the DTW algorithm. Finally, we compare percentage of correctly recognized sentences from the OSR with the percentage of correctly recognized sentences after DTW and mapping to one of the sentences of the corpus.

We used our modeling tool to create an own model for the execution of the evaluation. The model uses the same approach as described in the sections before. To collect the results a script function was added to the OsrCommand-Plugin to export into a csv file. Figure 4 shows the evaluation model during runtime. It calculates the statistics for the recognizer and the recognizer utterances which are processed by the algorithm. The participant starts a recognition of the given sentence by pressing the push to talk (*PTT*) button. If the recognition is finished the participant is able to display the next sentence by pressing the *Next* button. At the same time the statistics are updated.

7 Evaluation

For the evaluation we had 11 male and 11 female participants. None of the participants was a participant of the librispeech corpus collection. Most of the participants were non-native speakers. The evaluation was conducted in English. Most non-native speakers had a region specific pronunciation. To speak 50 sentences into the system took 15 minutes per participant. Table 5 shows the results of the evaluation. The usage of the DTW algorithm reduces the sentence error rate of each gender by absolute 31%. Our automotive domain specific data shows without DTW processing a lower sentence error rate than the librispeech test data in Table 4.

The usage of the DTW algorithm improves the sentence error rates of native and non-native speakers. The high sentence error rate of 78% could be lowered by the DTW algorithm to 14%. The six native speakers could also benefit of the usage of the DTW algorithm. The worst sentence error rate was 14% of one of the 16 non-native speakers.

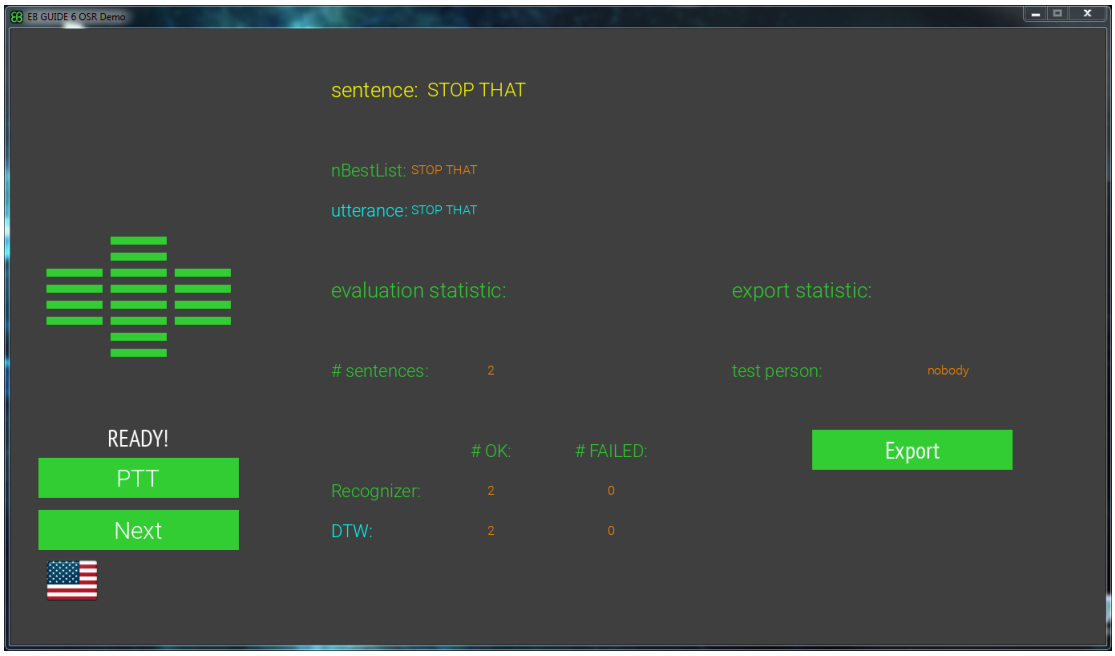

Figure 4 – Evaluation model during runtime. Buttons for the workflow are on the left side. The csv export options are on the right side. On the top the sentence is displayed and the first recognition result of the recognizer (nBestList) as well as the result after the processing of the DTW algorithm (utterance). The statistic shows the accumulated number of evaluated sentences (# sentences) and accumulated numbers for the recognizer and the DTW algorithm (# OK and # FAILED).

Table 5 – Evaluation of the sentence error rate (SER) for different speakers with the lbirispeech TDNN acoustic model and the `tgsmall` language model before and after applying of the DTW algorithm

Speaker	**SER**	
	TDNN	**TDNN + DTW**
female	34.18%	3.09%
male	34.55%	3.27%
summary	34.36%	3.18%
best native speaker	6.00%	0.00%
best non-native speaker	12.00%	0.00%
worst native speaker	18.00%	0.00%
worst non-native speaker	78.00%	14.00%

8 Conclusion

In this paper the Kaldi toolkit was evaluated for a command and control system in an automotive scenario. We trained a Kaldi model on the librispeech corpus and integrated a decoder to provide recognized utterances to the speech dialog system. Additionally we integrated the dynamic time warping (DTW) algorithm to reduce the sentence error rate by mapping recognized utterances onto intents. We trained different Kaldi models with the librispeech corpus for our recognizer. The best model based on TDNN had a word error rate of 4.15% on the test data with a four-gram language model, the model with the best size achieved 5.91% word error rate and 52.52% sentence error rate.

We evaluated the DTW algorithm using an EB GUIDE model together with our trained Kaldi TDNN model based on the small tri-gram language model. 22 people participated on our evaluation to speak 50 random sentences each. The algorithm reduced the sentence error rate from 34% to 3%. The usage of the DTW algorithm in the dialog system enables the mapping of spoken utterances onto intents in order to control an infotainment system. However, to

fully support all requirements of state-of-the-art automotive infotainment systems, additionally dynamic vocabulary like tuner stations or adressbook contacts need to be recognized and converted into intent parameters. For such use cases a named entity extraction needs to be added to our system as a next step.

References

[1] POVEY, D., A. GHOSHAL, G. BOULIANNE, L. BURGET, O. GLEMBEK, N. GOEL, M. HANNEMANN, P. MOTLICEK, Y. QIAN, P. SCHWARZ, J. SILOVSKY, G. STEMMER, and K. VESELY: *The kaldi speech recognition toolkit*. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, 2011. URL `http://publications.idiap.ch/downloads/papers/2012/Povey_ASRU2011_2011.pdf`.

[2] YE, H. and S. J. YOUNG: *A clustering approach to semantic decoding*. In *Proc. Interspeech 2006*. ISCA, 2006. URL `http://www.isca-speech.org/archive/interspeech_2006/i06_1118.html`.

[3] LEVENSHTEIN, V. I.: *Binary codes capable of correcting deletions, insertions and reversals*. *Doklady. Akademii Nauk SSSR*, 163(4), pp. 845–848, 1965. URL `http://ci.nii.ac.jp/naid/10024878029/en/`.

[4] TUR, G. and R. DE MORI: *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons, 2011.

[5] WEILHAMMER, K., P. KUMAR, V. SPRINGER, and D. MASSONIE: *Spoken language understanding in embedded systems*. In *Proc. Konferenz Elektronische Sprachsignalverarbeitung (ESSV)*, vol. 81 of *Studientexte zur Sprachkommunikation*, pp. 69 – 76. TUDpress, Dresden, Germany, 2016.

[6] MASSONIE, D., C. HACKER, and T. SOWA: *Modeling graphical and speech user interfaces with widgets and spidgets*. *ITG-Fachbericht: Speech Communication*, (252), 2014. URL `https://www.vde-verlag.de/proceedings-de/453640016.html`. VDE Verlag GmbH, Berlin/Offenbach, ISBN 978-3-8007-3640-9.

[7] PANAYOTOV, V., G. CHEN, D. POVEY, and S. KHUDANPUR: *Librispeech: An asr corpus based on public domain audio books*. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210. 2015. doi:10.1109/ICASSP.2015.7178964.

[8] POVEY, D., V. PEDDINTI, D. GALVEZ, P. GHAHREMANI, V. MANOHAR, X. NA, Y. WANG, and S. KHUDANPUR: *Purely sequence-trained neural networks for asr based on lattice-free mmi*. In *Proc. Interspeech 2016*, pp. 2751–2755. 2016. doi:10.21437/Interspeech.2016-595. URL `http://dx.doi.org/10.21437/Interspeech.2016-595`.

[9] GRAVES, A., S. FERNÁNDEZ, and J. SCHMIDHUBER: *Bidirectional lstm networks for improved phoneme classification and recognition*. *Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005*, 2005. doi:10.1007/11550907_126. URL `http://dx.doi.org/10.1007/11550907_126`.