2020 16th International Conference on
Control, Automation, Robotics and Vision (ICARCV)
December 13-15, 2020. Shenzhen, China

Speech recognition system for a service robot - a performance evaluation

Besim Alibegović
Faculty of Electrical Engineering
University of Tuzla
Tuzla, Bosnia and Herzegovina
besim.alibegovic@fet.ba

Naser Prljača
Faculty of Electrical Engineering
University of Tuzla
Tuzla, Bosnia and Herzegovina
naser.prljaca@untz.ba

Melanie Kimmel
IAV GmbH
Berlin, Germany
melanie.kimmel@iav.de

Matthias Schultalbers
IAV GmbH
Berlin, Germany
matthias.schultalbers@iav.de

***Abstract*—In this work we adapt and evaluate different solutions for automatic speech recognition (ASR) to be used as an HMI for the assistant robot. Two on-device solutions: Kaldi (DNN-HMM) and Mozilla's DeepSpeech (end-to-end), and three internet service APIs: IBM Watson, Microsoft Azure and Google Speech to Text are evaluated. The systems are adapted to the domain of robot commands and evaluated on a set of expected inputs. As the goal is to retain the ability to recognise general language, the systems are also evaluated on out of domain data.**

***Index Terms*—Speech recognition, ASR, WER, Kaldi, DeepSpeech, IBM Watson, Microsoft Azure, Google Cloud**

I. Introduction

Automatic speech recognition systems have been an established research topic for decades. Some great breakthroughs have happened in the last ten years, almost solely as a result of using neural networks in various ways. Today, two main approaches are being used: traditional DNN-HMM and more modern end-to-end approaches. There are a lot of implementations of these systems available to be used in an application of choice. Some of the implementations are open-sourced and run on-device, while others are running on remote servers and offer a high-level API as a black box service.

Implementing a custom on-device system requires having a trained model. The model can be trained on custom data and many data sets for training are available [1][2][3]. However, using a pre-trained model greatly reduces the implementation time, complexity and needed resources. In cases where ASR systems are used in an application with a known domain, the performance of the used models can be improved by providing an in-domain text corpus.

In this work, we aim to find the best system to be used in a speech based HMI for a service robot. We first evaluate the available solutions with their baseline models. Next, we adapt the models by providing the examples of expected input and evaluate them again. For evaluation, two data sets are generated and used separately: in-domain data set containing examples of commands, and general English data set. The performance of the systems is evaluated according to the word error rate (WER).

The research leading to these results was conducted in cooperation with IAV GmbH.

Two open-sourced, on-device solutions are chosen for comparison: Kaldi toolkit [4] implementing a hybrid DNN-HMM approach and Mozilla DeepSpeech [5] with an end-to-end CTC approach. Furthermore, three API-s from tech giants are also compared: IBM Watson [6], Microsoft Azure [7] and Google Speech-to-Text [8].

II. Used systems

A. Kaldi

Kaldi is a widely used on-device toolkit for ASR research. It provides different types of DNN-HMM architectures named nnet1, nnet2, nnet3 and special implementation of nnet3 called chain. Different recipes are included for training on standard datasets, and can be modified to use custom data to train a new model. However, some pre-trained models are already available on their website [9]. In this work, we use a pre-trained chain model made available by Günter Bartsch on Zamia project [10]. Its acoustic model is trained on over 1500 hours of audio data and it has a lexicon of 160k words.

Zamia project also provides a python wrapper for Kaldi which is used in this work to simplify the evaluation pipeline.

In this work the pre-trained model from Zamia is adapted by recompiling the HCLG graph with a custom language model and adding missing words to the vocabulary and pronunciation lexicon.

B. Mozilla DeepSpeech

DeepSpeech is an open source ASR system based on Baidu's research paper from 2014 [11]. It implements an end-to-end ASR approach with a CTC cost function and is implemented using Tensorflow.

Mozilla provides official pretrained models available for download. The English model is trained on 3816 hours of transcribed audio from Common Voice English [2], LibriSpeech [1], Fisher [3] and Switchboard [12] datasets. The model also includes around 1700 hours of transcribed radio shows.

C. Google Speech-to-Text

Google's ASR solution is called *Speech-to-Text* and supports more than 120 languages and variants. On the official

website [8], unparalleled accuracy as well as most advanced algorithms are claimed. Adaptation can be done by providing common text examples from a target domain during decoding.

Google also offers a Beta version of the API with some added features [13]. It provides added features such as automatic punctuation, speaker diarization, auto language detection and improved speech adaptation. In terms of adaptation, from the APIs point of view there is a new configuration variable called *boost* which can be used to scale adaptation up or down.

D. IBM Watson

IBM offers an ASR service as a part of Watson - their suite of AI services. Watson supports eleven languages at the moment including some that are still in beta [6]. It provides different base models for US and UK English and also provides a broad-band and narrow-band models for 16kHz and 8kHz audio respectively.

With a paid plan, Watson offers full adaptation capabilities which are comparable with custom local solutions such as Kaldi. The custom language or acoustic models need to be trained with users data in advance, but training only takes a few minutes. Customisation options include adding new words to the vocabulary, adding a new corpus to expand the language model and adding transcribed audio to customize the acoustic model. There is also an option to control the intensity of customisation through a parameter called *customisation weight*.

E. Microsoft

Microsoft also has a suite of cloud services named Azure. As a part of Azure's Cognitive Services they provide an API for speech recognition called Speech to Text. On the official website [7] it is claimed to use "breakthrough speech technology" powered by decades of research. Their website also refers to a paper from 2017; in which Microsoft's team achieved a first ever human level accuracy on Switchboard test [14].

However, adaptation is provided by a separate company *Scriptix* [15]. Scriptix company offers full adaptation of language and acoustic models through Microsoft Azure API.

III. Adaptation

Each of the evaluated systems is adapted by using an in-domain text corpus. For Kaldi, an in-domain corpus is doubled and combined with a general English corpus containing 2.5 million sentences. This combined corpus is used to train a new pruned 4-gram language model using KenLM package [16]. The same combined corpus is used to train a language model for DeepSpeech as well, but this time without pruning as the reduction in size was not necessary. Later, a new language model to be used in DeepSpeech is trained using only the in-domain adaptation corpus.

Each of the web services provides a way of adapting a model based on examples of in-domain text. For IBM Watson and Microsoft Azure API-s, that involves uploading the adaptation corpus and training a new model in advance. To adapt Google Speech-to-Text, examples of expected commands are provided during the upload of audio and no pre-training is done. However, the number of lines is limited to 5000, number of characters per line to 100 and total number of characters to 100,000. Therefore, a smaller adaptation set needed to be used for Google to fit into these constraints.

A. Adaptation corpus

The adaptation or in-domain dataset needs to resemble commands that can be given to the robot. The human interaction with the target Franka robot is limited by the functionalities it can provide. In the Franka robot functionality, several possible command intents are identified:

- Give an object
- Take an object
- Drop an object
- Move to home position
- Negative response or stop command
- Greeting a robot

Furthermore, commands with intents to give, take and drop can contain location information, item color or size e.g.:

> *please collect the box that is right of the white pliers.*
> *could you put the silver wrench thirteen on the floor.*

The robot can only take the objects if it can recognise them. In other words, the domain of objects the robot can manipulate is constrained by the implemented vision system. Similarly, the system has a domain of intents, objects, colors and sizes it can understand and handle. To generate a large number of these in-domain examples online dataset generator Chatito [17] is used. It generates data based on a set of rules. An example of a rule is:

```
\textildelow [please?] ~[give] ~[
   article?] ~[attribute?] ~[tool]
```

Where each of the terms in squared brackets is a name for the group of terms that can take that place. The *[give]* stands for the number of phrases or words with the same meaning - to give:

```
~[give]
    give ~[me?]
    i need
    pass ~[me?]
    hand over
```

The lists of 12 relevant items, 14 colors and 5 sizes, shown in Table. I, are defined similarly as well as other sentence forming parts such as attributes, articles, propositions etc.

Based on these rules a set of 60677 unique examples of commands is generated and used as an adaptation set.

IV. Evaluation set

When applying a speech recognition system on a specific task, in this case commands for a robot, a good evaluation

Items	Colors	Sizes
hammer	red	big
screwdriver	orange	small
pliers	yellow	bigger
tweezers	green	smaller
glass	blue	medium
box	purple	
calculator	brown	
pen	turquoise	
pencil	silver	
brush	violet	
wrench	pink	
	black	
	white	
	gray	

TABLE I
LIST OF ITEMS, COLORS AND SIZES DEFINED IN THE ADAPTATION SET.

set for system performance is one that contains examples of the expected input. Furthermore, it should be taken into account, who is going to be using the system and in what environment. For example, the dataset is going to be more suitable if the test speakers have the same accent as the target speaker. If the microphone of the target system is known and available it is best to use the same type during data collection. Otherwise, multiple different microphones should be used to minimise the effect of microphone quality on the results. For this work, we create two separate evaluation sets: in-domain evaluation set which contains examples of commands, and general English data set.

A. In-domain evaluation set

In-domain evaluation set consists of 573 recordings of different speakers giving 30 different commands. The commands have various length and complexity. Some examples are:

Please put the pliers near the box.
Give me the box that is left of the hammer.
Could you please hold this for a minute.
Move away.
Do not do that.

All speakers are non-native and have either German or Bosnian accent. In total there are 25 speakers of which 20 are male and 5 are female. Recordings are taken with different microphones including a number of Android and iPhone smartphones, different headsets, different laptop microphones and one conference speakerphone. All gathered files are converted to 16kHz PCM format which all compared systems natively support. Examples are recorded in the office, at home or in a café for some background noise.

B. General English evaluation set

The ASR systems are still expected to keep the ability to transcribe general English sentences after adaptation. To test that, a smaller general English data set is also generated consisting of 63 recordings of English sentences hand picked from news articles and dialogue examples found online. All of 11 speakers are male with Bosnian accent. Sentences range from shorter examples from a dialogue to standard size news sentences. Some examples are:

They hoped their conclusions would help to prepare astronauts for missions in space.
The central boulevard between the conference halls. has been transformed into a controlled zone.
Where are you going.
That sounds great.

V. RESULTS

Each system is evaluated first with the baseline model on both in-domain and general English datasets. Next, each system is customised using the adaptation corpus and evaluated again. The baseline results show what is to be expected from models without any modifications. They are also used to see the effects of adapting the system to the specific domain and also to see how much domain adaptation effects the general English performance. Finally the best version of each system is used to rank the solutions based on WER.

A. Kaldi

Since the performance of the Kaldi model is heavily influenced by the value of the acoustic scale, different values are tested. Different results for the Kaldi baseline model are shown in Table II. Beam width is held constant at the value of 60.

TABLE II
RESULTS FOR THE KALDI BASELINE MODEL.

Acoustic scale	In-domain WER(%)	General English WER(%)
0.9	14.58	8.85
1.0	**14.13**	**8.85**
1.1	14.06	9.07
1.3	14.33	8.85
1.5	15.03	9.07
1.7	15.86	10.58

After adapting with a new language model, the Kaldi model is evaluated again for various acoustic scale values. Adaptation results in a relative WER improvement of more than 90% on in-domain dataset, as seen in Table III.

TABLE III
RESULTS FOR THE KALDI ADAPTED MODEL

Acoustic scale	In-domain WER(%)	General English WER(%)
0.7	1.52	9.93
0.8	1.28	8.64
0.9	**1.32**	**7.56**
1.0	1.35	8.42
1.1	1.32	8.86
1.3	1.73	8.86
1.5	2.18	9.72
1.7	2.67	10.37

B. Mozilla DeepSpeech

DeepSpeech has three parameters that influence the performance during decoding: lm_alpha, lm_beta and beam width. Parameter *lm_alpha* is a language model weight, *lm_beta* is a word insertion penalty and *beam width* is the width of the pruning beam in the decoder.

The baseline model results with lm_beta = 1.85, beam width = 1000 and different values of lm_alpha are shown in Table IV.

TABLE IV
RESULTS FOR MOZILLA DEEPSPEECH BASELINE MODEL.

lm_alpha	In-domain WER(%)	General English WER(%)
0.5	24.45	24.84
0.75	22.33	22.68
1.5	25.48	22.68
3	59.73	44.71

A new language model is built based on the same corpus as for Kaldi - adaptation set copied twice and 2.5 million general English sentences. The results with this adapted language model are presented in Table V.

TABLE V
RESULTS FOR MOZILLA DEEPSPEECH ADAPTED MODEL

lm_alpha	In-domain WER(%)	General English WER(%)
0.4	26.83	25.05
0.75	20.36	20.95
1.5	25.41	23.33

Using the language model trained both on adaptation and general English corpora only slightly increased the performance. Therefore, a new language model is built, this time without the general English corpus to try to improve performance, at least on the in-domain data. This resulted in almost 50% relative improvement on in-domain data, which can be seen in Table VI. However, the general English WER suggests that this version of the system is unusable outside of the specified domain.

TABLE VI
RESULTS FOR MOZILLA'S DEEPSPEECH HEAVILY ADAPTED MODEL.

lm_alpha	In-domain WER(%)	General English WER(%)
0.4	14.37	80.99
0.75	12.22	75.59
1.5	11.98	80.13

C. IBM Watson

The results in Table VII show a big difference between Watson's baseline performance on in-domain and general data. The general English WER is almost 8% while the in-domain WER is almost 20%.

Models with different values of customisation weight are trained and evaluated and the results are shown in Table VIII.

TABLE VII
RESULTS FOR THE IBM WATSON BASELINE MODEL

IBM Watson	In-domain WER(%)	General English WER(%)
baseline	19.49	7.99

TABLE VIII
RESULTS FOR THE IBM WATSON ADAPTED MODELS

Custom. weight	In-domain WER(%)	General English WER(%)
0.6	2.73	9.07
0.7	2.56	9.93
0.8	2.28	10.58
0.9	2.08	11.66
1	5.19	90.06

D. Google

Google's baseline model achieves 13.8% WER on in-domain data and 7.5% on general English, as shown in Table IX. It does not provide any type of customization scaling in the official API. However, the Speech-to-Text Beta API has a boost variable which has similar function to customization weight in IBM Watson. Boost can have values between 0 and 20. However, the results with Beta version and Boost are worse than the standard Google API. Results are presented in Table IX and Table X.

TABLE IX
RESULTS FOR GOOGLE SPEECH-TO-TEXT BASELINE AND ADAPTED VERSIONS

	In-domain WER(%)	General English WER(%)
Baseline	13.85	7.56
Adapted	5.44	6.69

TABLE X
RESULTS FOR GOOGLE SPEECH-TO-TEXT BETA API

Boost	In-domain WER(%)	General English WER(%)
5	13.47	8.21
10	14.09	9.50
15	14.30	9.72

E. Microsoft Azure

Microsoft Azure's WERs are presented in Table XI. The baseline model shows great performance out of the box, achieving 6.67% WER on in-domain data and only 3.24% on general English. After training with the adaptation set, there is a relative WER improvement of almost 70% for the in-domain set compared to baseline model. The general English performance is unaffected by adaptation.

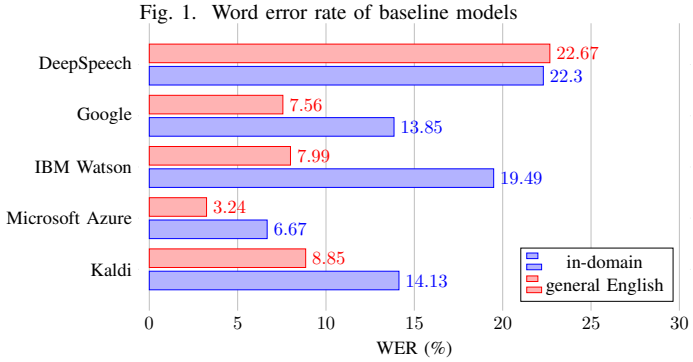
TABLE XI
RESULTS FOR MICROSOFT AZURE BASELINE AND ADAPTED MODELS

	In-domain WER(%)	General English WER(%)
Baseline	6.67	3.24
Adapted	2.15	3.24

F. Comparison of systems

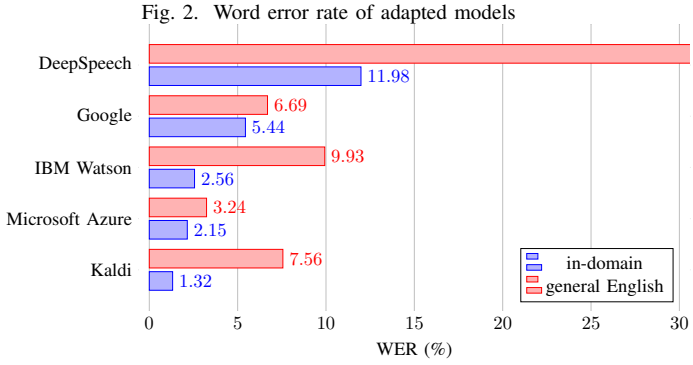
As seen in Fig. 1 the Microsoft Azure baseline model shows by far the best results both for general English and in-domain datasets. The Google, IBM and Kaldi baseline models perform similarly on general English while IBM is significantly worse on the in-domain dataset. Mozilla's DeepSpeech baseline model has the largest error on both datasets. The baseline models for Kaldi and DeepSpeech can be adjusted with parameters that determine the relative scaling of language and acoustic models, in order to favor one over the other. Only the best results are shown in Fig 1.

Fig. 1. Word error rate of baseline models



An overview of all adapted results ranked by performance on in-domain data is shown in Fig. 2. Kaldi achieves lowest WER in domain, followed by Microsoft's Azure which also has the best general English results.

Fig. 2. Word error rate of adapted models



VI. Conclusion

In this work we compared different speech recognition systems to find the best one to be used in the HMI interface of a robot. The results of the experiment suggest that Kaldi, when adapted with a custom language model, is more accurate than paid internet API services when transcribing the in-domain speech, and it has comparable accuracy on general English speech as well. In addition, as it is the on-device solution, it has the advantage of keeping all data private, no running cost, full flexibility, open-sourced code and no dependency on internet connection.

However, the results answer other more general questions. Mainly, they provide insights into the performance of currently available solutions for the recognition of general English speech. The experiment also reveals different impacts of adapting the systems to a specific domain and what accuracy is to be expected from adapted models.

The paper only evaluates the accuracy of the transcription while other important implementation aspects, such as latency or system requirements, are not considered. Also there are other ASR systems that can be evaluated in future work.

References

[1] V. Panayotov, 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)*, 2015, pp. 5206–5210.

[2] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, *Common voice: A massively-multilingual speech corpus*, 2019. arXiv: 1912.06670 [cs.CL].

[3] C. Cieri, D. Miller, and K. Walker, "The fisher corpus: A resource for the next generations of speech-to-text," in *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal: European Language Resources Association (ELRA), May 2004. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2004/pdf/767.pdf.

[4] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, IEEE Catalog No.: CFP11SRW-USB, Hilton Waikoloa Village, Big Island, Hawaii, US: IEEE Signal Processing Society, Dec. 2011.

[5] Mozilla. (accessed: 05.02.2020). Project deepspeech, [Online]. Available: https://github.com/mozilla/DeepSpeech.

[6] IBM. (accessed: 20.05.2020). Watson speech to text, [Online]. Available: https://cloud.ibm.com/docs/speech-to-text?topic=speech-to-text-about.

[7] Microsoft. (accessed: 20.05.2020). Azure speech to text, [Online]. Available: https://azure.microsoft.com/en-us/services/cognitive-services/speech-to-text/#features.

[8] Google. (accessed: 20.05.2020). Speech-to-text overview, [Online]. Available: https://cloud.google.com/speech-to-text.

[9] D. Povey. (accessed: 20.05.2020). Kaldi models, [Online]. Available: https://kaldi-asr.org/models.html.

[10] G. Bartsch. (accessed: 20.05.2020). English zamia speech model for kaldi, [Online]. Available: https://goofy.zamia.org/lm/2019/06/20/1500-Hours-160k-Words-English-Zamia-Speech-Models-Released.html.

[11] A. Y. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep speech: Scaling up end-to-end speech recognition," *CoRR*, vol. abs/1412.5567, 2014. arXiv: 1412.5567. [Online]. Available: http://arxiv.org/abs/1412.5567.

[12] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Proceedings of the 1992 IEEE International Conference on Acoustics, Speech and Signal Processing - Volume 1*, ser. ICASSP'92, San Francisco, California: IEEE Computer Society, 1992, pp. 517–520, ISBN: 0780305329.

[13] Google. (accessed: 20.05.2020). Speech-to-text beta, [Online]. Available: https://cloud.google.com/speech-to-text/docs/boost.

[14] W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke, "The microsoft 2017 conversational speech recognition system," *CoRR*, vol. abs/1708.06073, 2017. arXiv: 1708.06073. [Online]. Available: http://arxiv.org/abs/1708.06073.

[15] Scriptix. (accessed: 20.05.2020). Speech to text, [Online]. Available: https://www.scriptix.io/.

[16] K. Heafield, "Kenlm: Faster and smaller language model queries," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*, ser. WMT '11, Edinburgh, Scotland: Association for Computational Linguistics, 2011, pp. 187–197, ISBN: 9781937284121.

[17] R. Pimentel. (accessed: 20.05.2020). Chatito dataset generator, [Online]. Available: https://github.com/rodrigopivi/Chatito.