Automatic Speech Recognition in Taxi Call Service Systems

Samir Rustamov[1,2(✉)], Natavan Akhundova[3], and Alakbar Valizada[3]

[1] ADA University, Baku, AZ 1008, Azerbaijan
srustamov@ada.edu.az
[2] Institute of Control Systems, Baku, Azerbaijan
[3] ATL Tech, Baku, AZ 1022, Azerbaijan
{natavan.akhundova,alakbar.valizada}@atltech.az

Abstract. In this research, the application of automatic speech recognition system in taxi call services is investigated. In comparison with traditional query handling systems such as live agents, Interactive Voice Response systems, type-base websites and mobile applications, the newest trend of artificial intelligence - speech recognition can be applied to make conversations in more natural way. For developing, training and testing of the system, Kaldi and CMUSphinx open-source speech recognition tools were utilized. Approximately 4 h of speech data in Azerbaijani have been processed for both tools. Testing has been accomplished in two ways; one of which is recognizing dataset from unknown speakers, and the other one is recognizing shuffled dataset. During these tests, variance and speed were investigated, along with accuracy. Kaldi showed accuracy between 97.3 and 99.6 with variance changing between 0.03 and 4.8. On the other hand, CMUSphinx attained accuracy between 95.6 and 97.8 with variance values of 0.2 and 3.8 in relatively less training time. Accomplished results were compared and used to define appropriate parameters for investigated models.

Keywords: Speech recognition · Kaldi · CMUSphinx · n-gram · Taxi call service · Speech features

1 Introduction

In the hectic life of big cities where people are always in rush, taxis play a valuable and crucial role. As demand increases, taxi companies see an incentive to improve quality of the service, as well as, enhancing query handling methods in call centers. Nevertheless, there is a still problem of customers waiting in long queues, especially, in peak hours. While call center agents work 24/7 to solve this incompetence, they are left with few or no social life. On the other hand, hiring more agents is a short-term solution for a company. The problem lies in the fact that a gap between the improvement of the service and today's technology exists. In order to eliminate this gap, most of the companies keep up with the latest trends of technology. Some made app and online websites, while others valued conversation more and refined calls with Interactive Voice Response (IVR) systems.

M. H. Miraz et al. (Eds.): iCETiC 2019, LNICST 285, pp. 243–253, 2019.
https://doi.org/10.1007/978-3-030-23943-5_18

All of these methods have its own pros and cons, however, this paper will focus on improvement of handling queries via call. By applying IVR to calls, certainly more queries can be handled parallelly without a human factor. The disadvantage of this method is that it quickly annoys customers and even can take more time than talking to a person because of starting over when pressed numbers wrongly.

In comparison with IVR systems, the newest trend of artificial intelligence - speech recognition can be applied to make conversations more natural as it does not limit customers with predefined options to choose. Speech recognition transforms speech to text while listening to a voice, and this is why it speeds up recording of an order, whereas previously agents needed to write it down. Having an automatic machine on 24/7 to handle customer queries can play as a competitive advantage for a company resulting in high revenue and customer satisfaction.

This paper promotes speech recognition as a tool for the use of taxi call centers in Azerbaijani language. It compares two open source tool kits - Kaldi and CMUSphinx for speech recognition with data in that language and discusses their advantage and disadvantages relative to the usability by call centers. Accordingly, Sect. 2 presents literature review, Sect. 3 is for an overview of speech recognition process, Sect. 4 introduces speech recognition using Kaldi and CMUSphinx, Sect. 5 delivers experimental results and Sect. 6 is about the discussion followed by the conclusion.

2 Literature Review

In [1], Matarneh, Maksymova, Lyashenko and Belova compared different close-source and open-source speech recognition tools based on various parameters, such as error rate, speed, response time and API. Authors tested Dragon Mobile SDK, Google Speech Recognition API, Siri, Yandex SpeechKit and Microsoft Speech API for close-source, and CMUSphinx, Kaldi, Julius, HTK, iAtros, RWTH ASR and Simon for open-source tools.

In [2], the statement that speech recognition technology has reached human performance by Microsoft is put under a test. Authors concluded that according to the test results, the statement being wrong is claimed.

Authors of [3] evaluated accuracies of three open-source toolkits: HTK, CMUSphinx and Kaldi based on German and English data. Based on their results, Kaldi outperformed the other tools.

In [4], authors integrate PyTorch, a library for neural network, and Kaldi, an open-source speech recognition toolkit, to obtain more efficient and accurate results. The authors confirmed their hypothesis via experiments with various datasets.

Parthasarathi and Strom in [5] build acoustic model with 7000 h labeled and 1 million hours of unlabeled data and discuss their results. The authors put forward the significance of data volume on recognition and how hyper-parameter tuning can improve accuracy.

Authors in [6] introduce a new system for recognizing a specific speaker in a signal with multiple speakers via training two neural networks. The system met expectations by increasing accuracy of recognition on both multi-speaker and single-speaker signals.

Schatz and Feldman discussed in [7] if one of the key parts of speech recognition - Hidden Markov Model or neural networks is more similar to human behavior and

perception of speech, via testing on corpuses in American English and Japanese. They concluded that neural networks have the best understanding of human perception of speech.

Fukuda et al. in [8], emphasized the problem of speech recognition on accepted speech and introduced data augmentation as a method to solve it. The authors modified accented data with three operations which are voice transformation, noise addition, and speed modification and concluded the last one being the most effective.

Jain, Upreti, Jyothi also referred to accented speech problem in [9] and suggested an architecture that learns a multi-accent acoustic model and an accent classifier. Together with speech augmentation, these techniques improved the performance of accent recognition.

In [10] authors experimented on recognition of speech on broadcasts when training data is scarce. The proposed approach was to collect data via related web sources in order to span different and broad domains.

In [11] Rustamov et al. developed speech recognition system for Flight Simulator Cockpit in C# from scratch which performed training with neural networks. In comparison with Microsoft Speech SDK, the tool achieved better results.

The idea of applying speech recognition technology in call centers is not new, and it has already been adopted by some companies in the beginning of a century. Australian company named "Regent Taxis" has implemented such a solution for its call center back in 2000 [12]. Within a few months, positive results have been achieved as automated technology gain popularity among users [13]. A company in New Zealand called "Co-op Taxis" was inspired by that and started to apply the same technology by the same vendor in 2001 [14].

In [15–19], authors applied different combination of hybrid neural networks for Azerbaijani isolated speech recognition systems.

3 Overview of Speech Recognition Process

Speech recognition is not a new term; it was first mentioned in 1930s [20]. Speech recognition, also known as speech-to-text, is a process of turning audio waves into texts. According to Jurafsky and Martin [21], in order to automatically recognize speech, four dimensions are considered: vocabulary, naturality of speech, noise and accent of a speaker. To begin with, the task of recognition becomes easier when vocabulary to be defined is small. Also, words isolated with pauses make the recognition easier, in comparison with continuous speech. This is because transforming continuous speech into text requires an additional tough task as separating speech to words, and this process can lead to errors. Finally, any kind of noise and accent in speech can decrease the accuracy of recognition due to the fact that such speeches do not come align with what the tool was trained.

The process of recognition is mainly based on probability and search. Given an audio input, estimates are defined for possible outputs, and an output with the highest

probability is searched. If we define sentences with W, the desirable output with $\grave{W}$, and the audio input with O, then our output can be expressed by

$$\grave{W} = \arg\max_{W \in L} P(W|O) \tag{1}$$

where L is a vocabulary in a given language. It means that, with the given audio input O, we take a sentence out of all sentences W which has maximum probability. By using Bayes' rule, this expression can be changed as

$$\grave{W} = \arg\max_{W \in L} P(O|W)P(W)P(O). \tag{2}$$

During calculation, P(O) does not have any effect on the result because it is the same for each value. We can rewrite the equation as

$$\grave{W} = \arg\max_{W \in L} P(O|W)P(W) \tag{3}$$

which gives us the product of acoustic model P(O|W) and language model P(W).

Language model is a set of probabilities for word sequences in the given language. The length of the sequence is defined by n-grams where n changes as 1, 2, 3 and et cetera. On the other hand, the acoustic model calculates probabilities of phones generating feature vectors at each time frame of audio. Feature vectors are vectors which include information about each time frame. The overall process of speech recognition is described in Fig. 1.

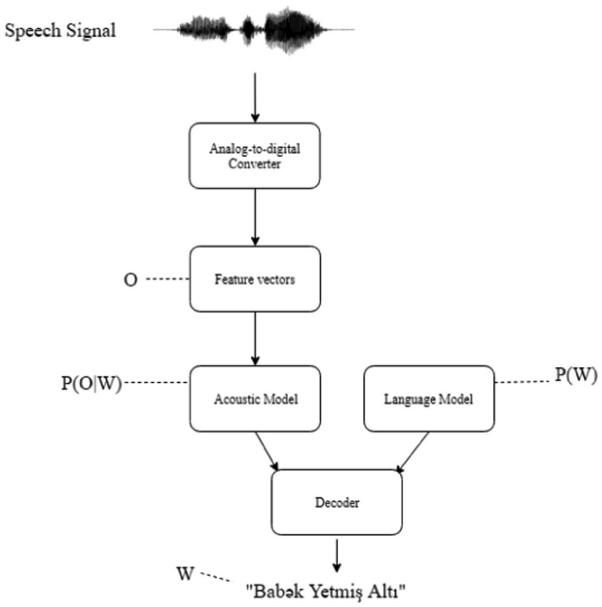

Fig. 1. Speech recognition process

4 Speech Recognition Using Kaldi and CMUSphinx

4.1 Overview of Toolkits

In order to build a speech recognition technology in any natural language, open-source tools will be needed. According to different blogs' top ratings such as LinuxLinks [22] and Silicon Valley Data Science [23], two of the most popular speech recognition tools are agreed to be Kaldi and CMUSphinx, also known as just Sphinx.

To begin with, Kaldi [24] is designed and intended for researchers on the field of speech recognition. It has been compiled on Windows, Linux, MacOs and is continuously updated by a lot of contributors. Therefore, the tool is full of various useful scripts and codes for any appropriate purpose. Currently, Kaldi is developing a new architecture for recognition which is deep neural networks.

CMUSphinx [25], on the other hand, is relatively easy to start with and has a simple documentation. It also has a huge community and in constant update. The toolkit consists of four parts: Pocketsphinx, Sphinxbase, Sphinx4 and Sphinxtrain. These parts are responsible for recognition, its lightweight and supporting libraries and training tools. Sphinx can be compiled on both Windows, Linux and MacOS.

Even though Kaldi was written on C++, while Sphinx on Java, both of the tools have been developed in such a structure where any new module can be easily added or removed [1]. The most accurate out of these two is Kaldi, [3] however Sphinx also will be tested in this paper to compare the results and other performances.

4.2 Data Preparation

The audio data used in this research is in the Azerbaijani language and equals approximately 3.52 h with 152 vocabulary words.

The data contains the most popular 100 addresses of the capital city of Azerbaijan, Baku. There also exist numbers in the names of streets, ranging from one till thousand. The shortest utterance consists of two words, whereas, the longest has eleven words. Examples for the data are: "ABBAS MİRZƏ ŞƏRİFZADƏ OTUZ DOQQUZ" - Abbas Mirza Sharifzada thirty nine; "FÜZULİ KÜÇƏSİ BİR" - Fuzuli street one; "VAQİF PROSPEKTİ" - prospect of Vagif and et cetera.

The audio recordings were recorded via ordinary microphones at 16 kHz with minimal noise and accurate grammar and pronunciation. Speakers, with total number of 62, were students within the age range from 18 till 24. fluent, which means no dialects were used during speaking. Nevertheless, database comprises different speaking styles and tonalities. Each subject pronounced all street names once. Some of the recordings have been removed due to speaker or microphone error, leaving, in total, 6121 utterances. The data was trained as continuous speech.

Data processing for both tools were nearly the same. Sphinx and Kaldi require defining utterance ids, a map of utterance ids to transcriptions and a dictionary with phonetic transcriptions. Additionally, Kaldi needs speaker information like gender and a map of speakers and utterance ids.

When it comes to language model, Kaldi has internal scripts for creating an n-gram, and when the system is put to be trained, an n-gram will be automatically created. To

train CMUSphinx based system, an additional command should be executed to create an n-gram, beforehand.

Overall, CMUSphinx has less steps in configuration for training a dataset than Kaldi. This is because Sphinx is aimed at developing practical applications [25], whilst, Kaldi is for researches.

5 Experimental Results

5.1 Overview of Experiments

The experiments comprise of checking accuracies of training and testing on audio data, which are assumed to be useful for taxi call systems. The trainings were conducted using both open-source speech recognition tool kits: Kaldi and CMUSphinx. The environment was the same virtual platform in order to eliminate bias which could have possibly occurred during training and testing the tools. The OS was Kali Linux and was running using single i7 CPU at 1.80 GHz speed.

Accuracy of training is carried out by testing all data against training set, however for calculating the accuracy of tests only ten percent of all data was considered as test, exclusively.

To begin with, different n-grams were used while the trainings, starting from 1 till 5, in order to know how accuracies are changing. This experiment will be conditionally named as "100/100". Both its training and test sets equally have 6121 utterances. The last two tests were performed ten times with random ten percent of data and with 3-grams as a language model. These tests will be named: "90/10 speaker" and "90/10 shuffle". In "90/10 speaker" test, ten percent of all speakers were exclusively given as a test and the rest for a training set. The size of testing set varies within a range of 598 and 612 utterances due to the fact that speakers have different number of recordings, and excluding some accounts for a difference in count. Finally, for conducting "90/10 shuffle" test, ten percent of all data, which is 612 utterances, was excluded from overall database as a testing set.

Accuracies of experiments are defined by Word Error Rate (WER) and Sentence Error Rate (SER). WER is a ratio of inserted (I), deleted (D) and substituted words (S) to the amount of all words (N) within given audio input [21]. It is calculated as

$$WER = \frac{S + D + I}{N} \tag{4}$$

SER is a ratio of errors made while recognizing sentences to all sentences.

$$SER = \frac{E}{N} \tag{5}$$

In this formula, E stands for all sentences with substituted, deleted or inserted words, whereas N indicates all sentences [26].

5.2 Kaldi Results

On Kaldi ASR, training accuracy of "100/100" dataset starts at 3.03% for SER with 1-gram and rapidly decreases afterwards. Even though for WER changes are not so drastic, it can be observed that both error rates decrease till 3-grams and, after, remains stable at 0.36% for SER and 0.1% for WER (Fig. 2).

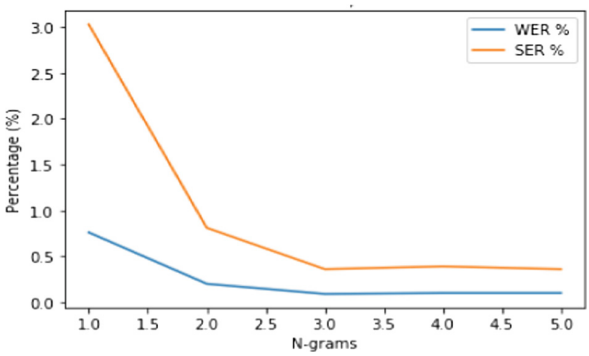

Fig. 2. Kaldi 100/100

Based on the results of training accuracy, the least error prone n-gram, which is 3-grams, was chosen for speaker and shuffle testing.

These tests showed accuracy rates between 97.3% and 99.6%, which are reasonably high results. Variances are all below 1, except for SER in speaker testing. For Kaldi, it took 13 min and 19 s on average to finish a training on the indicted computer above (Table 1).

Table 1. Kaldi test descriptions

	WER %	Variance of WER	SER %	Variance of SER
90/10 speaker	1.020	0.884	2.699	4.750
90/10 shuffle	0.368	0.025	1.363	0.270

5.3 CMUSphinx Results

The next tool, CMUSphinx, starts off with relatively high values: 25.4% for SER and 7.6% for WER. It reaches stability again on 3-grams with "100/100" testing, at 3.7% for SER and 2.1% for WER (Fig. 3).

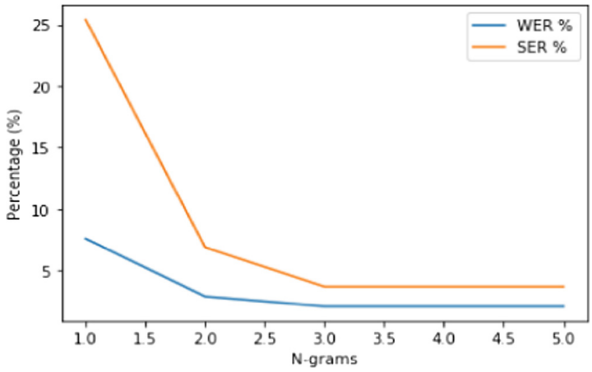

Fig. 3. CMUSphinx 100/100

For "90/10" tests, accuracy changes in between 95.6% and 97.8%, which means WER and SER for both tests are approximately 2% and 4%, respectively. During speaker testing, variance showed 2.147 for WER and 3.821 for SER. The variance is the highest among all tests due to one irregular result. Shuffle testing gave slightly better results, in comparison with the previous one. Sphinx, on average, finished a training in 9 min and 41 s on the indicated computer above (Table 2).

Table 2. CMUSphinx test descriptions

	WER %	Variance of WER	SER %	Variance of SER
90/10 speaker	2.200	2.147	4.350	3.821
90/10 shuffle	2.230	0.227	4.150	0.565

6 Discussion

After performing "90/10 shuffle" and "90/10 speaker" tests 10 times for each, together with testing different n-grams, the following results were obtained.

The most accurate n-grams for the current audio data starts from 3-grams. 3-grams are a common n-gram model for most languages, and based on the results, both tools match on that.

However, differences exist in experimental test results. Kaldi shows better performance on accuracy rates. The fact that Kaldi by far is the most accurate open-source toolkit is claimed also by other sources, including [3]. For variance of WER, Kaldi showed results below 1, whereas, Sphinx attained more than 2 on speaker testing. In terms of SER, variance was higher for both tools on speaker testing, however, Kaldi presented the highest. Furthermore, one of main advantages that CMUSphinx has on above-mentioned results is that the tool performed faster on the same dataset in training process than Kaldi (Table 3).

Table 3. Kaldi and CMUSphinx comparative test results

	Kaldi		CMUSphinx	
	90/10 speaker	90/10 shuffle	90/10 speaker	90/10 shuffle
WER%	1.020	0.368	2.200	2.230
SER%	2.699	1.363	4.350	4.150
Variance of WER	0.884	0.025	2.147	0.227
Variance of SER	4.750	0.270	3.821	0.565

Among all tests, shuffle showed better results than speaker testing. This is because while shuffle testing, tools do not attempt to recognize the voices they were not trained with. All in all, the most accurate result was gained on Kaldi shuffle testing, whereas, the least accurate test was speaker testing on CMUSphinx. Tools perform differently on variance results for WER and SER. Regarding speed, CMUSphinx finished faster than Kaldi.

7 Conclusions

This research paper investigated the application of speech recognition open-source toolkits on taxi call service systems. The toolkits - Kaldi and CMUSphinx were used to train and test a dataset of almost four hours.

The dataset, comprised of 100 addresses in Azerbaijani, was put into "90/10 speaker" and "90/10 shuffle" tests. Additionally, tools were examined in terms of accuracy with different n-grams. Tools coincide with results of n-gram testing, but differ on other two. Kaldi showed lower word and sentence error rates than Sphinx on speaker and shuffle tests. The lowest variances of WER and SER values belongs to Kaldi, however, Kaldi also holds the highest value for variances of SER. Sphinx presented the highest variance in WER values and moderate variance in SER values. It is worthy to note that, with the given hardware, CMUSphinx finished trainings faster than Kaldi.

With given results it could be concluded that for obtaining fast results CMUSphinx can be easy to configure and fast to train. Nevertheless, in order to have an accurate tool with relatively low level of variance, Kaldi should be chosen. Taxi companies will need to consume time and money for applying speech recognition systems to call centers, but will eventually gain return on investment with such an accurate tool as Kaldi.

Acknowledgment. This work has been carried out in Center for Data Analytics Research at ADA University and in Research and Development Laboratory at ATL Tech.

References

1. Matarneh, R., Maksymova, S., Lyashenko, V.V., Belova, N.V.: Speech recognition systems: a comparative review. IOSR J. Comput. Eng. **19**(5), 71–79 (2017). https://www.researchgate.net/publication/320673436_Speech_Recognition_Systems_A_Comparative_Review
2. Saon, G., et al.: English conversational telephone speech recognition by humans and machines, March 2017. https://arxiv.org/pdf/1703.02136v1.pdf

3. Gaida, C., Lange, P., Petrick, R., Proba, P., Malatawy, A., Suendermann-Oeft, D.: Comparing open-source speech recognition toolkits. In: 11th International Workshop on Natural Language Processing and Cognitive Science (2014). http://suendermann.com/su/pdf/oasis2014.pdf
4. Ravanelli, M., Parcollet, T., Bengio, Y.: The pytorch-kaldi speech recognition toolkit, February 2019. https://arxiv.org/pdf/1811.07453v2.pdf
5. Parthasarathi, S.H.K., Strom, N.: Lessons from building acoustic models with a million hours of speech, April 2019. https://arxiv.org/pdf/1904.01624.pdf
6. Wang, Q., et al.: VoiceFilter: targeted voice separation by speaker-conditioned spectrogram masking, February 2019. https://arxiv.org/pdf/1810.04826v4.pdf
7. Schatz, T., Feldman, N.H.: Neural network vs. HMM speech recognition systems as models of human cross-linguistic phonetic perception. In: 2018 Conference on Cognitive Computational Neuroscience (2018). http://thomas.schatz.cogserver.net/wp-content/uploads/2018/11/Schatz2018b.pdf
8. Fukuda, T., et al.: Data augmentation improves recognition of foreign accented speech. Interspeech, September 2018. https://www.isca-speech.org/archive/Interspeech_2018/pdfs/1211.pdf
9. Jain, A., Upreti, M., Jyothi, P.: Improved accented speech recognition using accent embeddings and multi-task learning. Interspeech, September 2018. https://www.isca-speech.org/archive/Interspeech_2018/pdfs/1864.pdf
10. Ragni, A., Upreti, M., Gales, M.J.F.: Automatic speech recognition system development in the "Wild". Interspeech, September 2018. https://www.isca-speech.org/archive/Interspeech_2018/pdfs/1085.pdf
11. Rustamov, S., Gasimov, E., Hasanov, R., Jahangirli, S., Mustafayev, E., Usikov, D.: Speech recognition in flight simulator. aegean international textile and advanced engineering conference. IOP Conf. Ser. Mater. Sci. Eng. **459** (2018). https://iopscience.iop.org/article/10.1088/1757-899X/459/1/012005/pdf
12. Forsyth, A.: Taxi Company Adopts Speech Recognition Technology. Computerworld, 26 October 2000
13. Forsyth, A.: Taxi fleet bets on speech recognition, Computerworld, 18 May 2001
14. Malcolm, A.: Cab firm books speech recognition system, Computerworld, 17 May 2001
15. Aida-Zade, K., Ardil, C., Rustamov, S.: Investigation of combined use of MFCC and LPC Features in Speech Recognition Systems. World Acad. Sci. Eng. Technol. Int. J. Comput. Inf. Eng. **1**, 2647–2653 (2007)
16. Aida-Zade, K., Rustamov, S.: The principles of construction of the azerbaijan speech recognition system. In: The 2nd International Conference "Problems of Cybernetics and Informatics", pp. 183–186 (2008)
17. Aida-Zade, K., Rustamov, S., Mustafayev, E.: Principles of construction of speech recognition system by the example of azerbaijan language. In: International Symposium on Innovations in Intelligent Systems and Applications, pp. 378–382 (2009)
18. Ayda-zade, K., Rustamov, S.: Research of cepstral coefficients for azerbaijan speech recognition system. Trans. Azerbaijan Natl. Acad. Sci. Inform. Control. Probl. **3**, 89–94 (2005)
19. Aida-zade, K., Xocayev, A., Rustamov, S.: Speech recognition using support vector machines. In: 10th IEEE International Conference on Application of Information and Communication Technologies, AICT 2016 (2016)
20. Juang, B.H., Lawrence, R.: Automatic Speech Recognition - A Brief History of the Technology Development, January 2005
21. Jurafsky, D., Martin, J.H.: Automatic speech recognition. In: Speech and Language Processing, pp. 285–291. Pearson Education (2008)

22. Emms, S.: Best Free Linux Speech Recognition Tools – Open Source Software, LinuxLinks 3 March 2018
23. Thompson, C.: Open Source Toolkits for Speech Recognition, Silicon Valley Data Science, 23 February 2017
24. Kaldi ASR. kaldi-asr.org. Accessed 16 April 2019
25. CMUSphinx Open Source Speech Recognition. cmusphinx.github.io. Accessed 16 Apr 2019
26. Bagiyev, A., Gurbanli, K., Mammadova, N., Nuriyeva, S.: Development of limited-vocabulary ASR for Azerbaijani. ACM Celebration of Women in Computing womENcourage 2018, October 2018. https://womencourage.acm.org/2018/wp-content/uploads/2018/07/womENcourage_2018_paper_26.pdf