American Journal of Neural Networks and Applications
2021; 7(1): 15-22
http://www.sciencepublishinggroup.com/j/ajnna
doi: 10.11648/j.ajnna.20210701.13
ISSN: 2469-7400 (Print); ISSN: 2469-7419 (Online)

Early Stages of Automatic Speech Recognition (ASR) in Non-english Speaking Countries and Factors That Affect the Recognition Process

Dunya Yousufzai

Computer Science, Software Engineering Department, Kabul University, Kabul, Afghanistan

Email address:
dunyayousufzai@gmail.com

To cite this article:
Dunya Yousufzai. Early Stages of Automatic Speech Recognition (ASR) in Non-english Speaking Countries and Factors That Affect the Recognition Process. *American Journal of Neural Networks and Applications*. Vol. 7, No. 1, 2021, pp. 15-22.
doi: 10.11648/j.ajnna.20210701.13

Received: April 22, 2021; **Accepted**: May 17, 2021; **Published**: May 31, 2021

Abstract: There has been a considerable stream in ASR over the past few decades, but it may seem strange why this field is still a subject for researchers to work on. There are many reasons, but somewhat because the discipline is created with the promise of human-level performance under pragmatic states and this is an inextricable problem. In addition, the increasing advancement of technology in various fields has caused a more compelling need for this field. Especially the establishment of such a system in the security sector in insecure third world countries such as Afghanistan is an urgent need. This paper began with the reflection of all the necessary knowledge about speech recognition and then suggested an unprecedented method for building an automated speech recognition (ASR) system in the Dari language using the two most powerful open source engines CMUSphinx, from Carnegie Mellon University and DeepSpeech v0.9.3 /. These systems are much more impressive than early speech recognition systems. Using my own collected dataset, a speech-to-text model has been trained for the Dari language. Firstly, the dataset is filtered according to the task, then demonstrated the possible compatibility from the hidden Markov (HMM) models, the phoneme concept to RNN training. The system surpassed previously predicted results, as CMUSphinx stated, “for a typical 10-hour operation, the WER should be around 10%." Finally, 3.3% WER was achieved with 10.3-hours of audio recording using CMUSphinx. 1% WER with DeepSpeech.

Keywords: Dari Language, HMM (S), Neural Network, Non-speaking English Countries, RNN, Speech-recognition, WER

1. Introduction

From prehistoric times until now, the exchange of information and considerable effort in interlocution has been and will be a considerable purpose to improve human understanding, so that hearing twiddles an important role in this process. Hearing hinges on a series of complex and intricate stages which convert sound waves in the air into electrical signals. The brain receives these signals with the help of the auditory nerve. Sound waves arrive in the outer ear and pass through a narrow canal called the ear canal, which conducts to the eardrum. As the sound waves enter, the eardrum starts vibrating. There are three tiny bones in the middle ear called the malleus, incus, and stapes that receive the vibration. The role of these bones in the middle ear is to amplify the sound vibrations and send them to the snail-shaped structure called the cochlea. Cochlea filled with fluid. When the fluid inside the cochlea twist and turns, a moving wave will be generated along the basal membrane. Hair cells - Sensory cells located above the basal membrane. The ions reach the top of the cell and secrete chemicals at the bottom called neurotransmitters. These chemicals attach to the auditory nerve and produce an electrical signal that eventually travels to the brain [1]. As you can see, comprehension and hearing are different at the human level, but this is where unprecedented achievements in SR (speech recognition) comes in. Automatic Speech Recognition (ASR) is a technology that allows a computer to identify the words that a person speaks into a microphone or telephone [2]. The wide availability of devices equipped with microphones and powerful computing capabilities constitutes a great potential for using ASR systems [3].

In recent years, there have been significant advances in machine learning algorithms, which have provided the basis for the development of various handy applications. Deep learning uses cross-sectional studies that help applications such as speech recognition. It should be mentioned that different types of neural networks play a key role in the field of ASR by being at the core of machine learning. There has been comparatively enough recognition research on the Persian language contrasted to the Afghanistan Dari language, although Dari is the main language but unfortunately no one has worked in this field so far. So in this article, a thorough discussion about the steps of manual engineering processing, extremely modular and pliable ways which have support for a variety of HMM-based acoustic models, about numerous language models and probe tactics with respect to CMUSphinx and an end-to-end speech system has been made. By following these steps, you can build a model for your language that is still unpopular and unknown. Meanwhile, alleges disparate challenges: (i) a shrewd track to collect a large number of the dataset and filter it to productively put upon all of them must be found, (ii) it is necessary to be familiar and have enough information in this field, (iii) it is compulsory to have sufficient ability to switch between different types of algorithms and frameworks according to your needs (iv) and if you want to work with phonemes, you must create a phonetic dictionary for your own language even no one has worked on it yet. In the continuation of this article, the concept of a speech recognition system will be discussed. It begins by describing the basic complexity of the neural network and the process of training your own dataset with hard encryption, starting from scratch in Section 2, followed by a discussion on CMU Sphinx and how to prepare your own dataset (Section 3). And move on to the recurrent neural network (RNN), talking about DeepSpeech and preparing a dataset for this framework (Section 4). And last it concludes with experimental results (Section 5), followed by conclusions.

2. Speech Recognition with CNN

Let's quickly find out what sound is? Sound is a longitudinal pressure wave composed of the impaction and dilution of air molecules, in a path equal to the application of energy. The areas where air molecules are forced to use energy to a more precise configuration than usual called impaction, and dilution are areas where air molecules are less packaged. The speed of a sound pressure wave in air is almost 331.5 + 0.6 Tc m/s, where Tc is the temperature of Celsius [Spoken Language Processing, 2001]. Here we have to convert the analog signal to a digital signal, which is a segregated exhibition of a signal over a period of time. So the kernel or core of speech-to-text conversion is the elicitation of various characteristics of the audio signal. Any physical element which is constant or variable in time is called a signal [4].

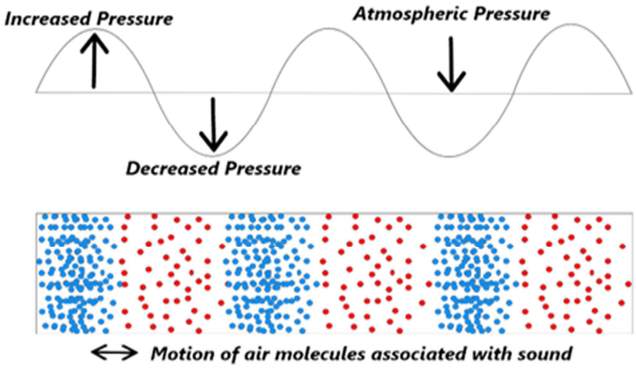

Figure 1. *Rarefaction and compression of air molecules.*

Convolutional neural networks (CNN) are the inspiration in the deep learning assembly. CNN utilizes a particular network frame, which is composed of intermittently called convolution and pooling layers. In CNN the input data required to be formed as multiple feature maps, which means it needs to organize speech feature vectors into feature maps [5]. The CNN exploits domain knowledge about feature invariances within its structure [6]. Recently CNN has met a significant research progress [7-9]. A convolutional neural network consists of an input layer, hidden layers, and an output layer. The dot product is done by the hidden layer as the hidden layer is responsible to perform convolutions.

$$a \cdot b=\sum_{i=1}^{n} a_i\, b_i=a_1\, b_1 + a_2\, b_2 + a_3\, b_3 + \ldots.+ a_n\, b_n$$

and its activation function is commonly Rectifier (ReLU), F(x)=max (0, x), A clear approach to the rectifier is the analytic function f(x)=ln (1 + ex).

The convolution operation is a linear operation, demonstrated by an asterisk, that consolidates two signals.

$$f[x, y] * g[x, y]=\sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} f[n_1, n_2] \cdot g[x - n_1, y - n_2]$$

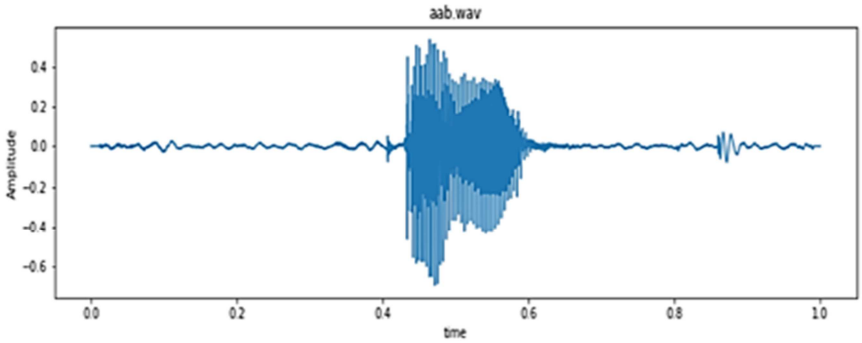

Figure 2. *Projection of Audio signal in time series domain.*

Here in CNN the input with a shape of N_i x I_h x I_w + I_c passing through a convolution layer. Convolutional networks may include local and/or global pooling which Pooling layers deduct the dimensions of data. Figure 2 demonstrates the projection of Audio signal in the time series domain, as 297,482 one-second recorded words (82.3 hours) have been put into specific folders so it is necessary to know the number of recordings for each voice command.

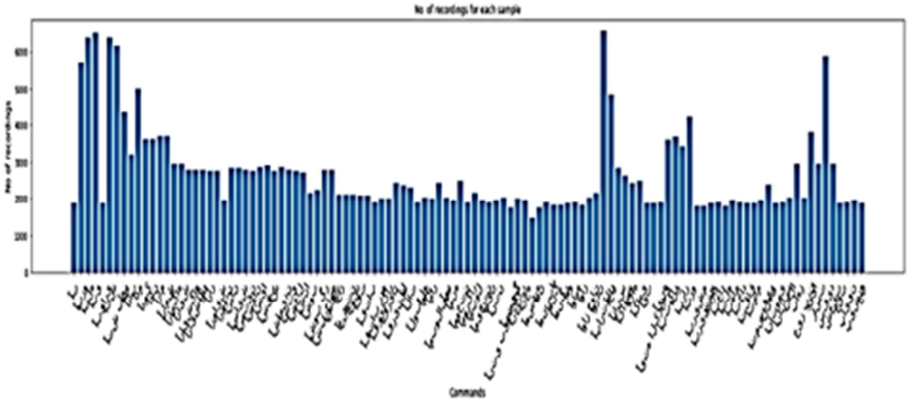

Figure 3. *The number of recordings for each voice command.*

Two steps more for resampling and removing shorter commands of less than 1-second have been followed. All of the labels and all of the waves have been extracted in order to get output labels, then converted the output labels to integer encoded, chasing this conversion of the integer encoded Labels to a one-hot vector took place because it is a multi-classification problem. Afterward, reshaped the 2D array to 3D since the input to the conv1d has to be a 3D array.

[[[5.6007931e-16]
[7.9060451e-16]
[1.4276164e-15]]]

The model has been trained on 80% of the data and validated on the remaining 20% then the speech-to-text model has been built-in using conv1d. Conv1d is a convolutional neural network that carries out the convolution along one dimension. For model building, Keras functional API is preferred and used Adam for optimizer and categorical cross-entropy for the loss, before long early stopping and model checkpoints for the callbacks to stop training the neural network at the right time has been used so that it gives the possibility to save the best model after every epoch, and finally, the data on batch size 32 has been trained.

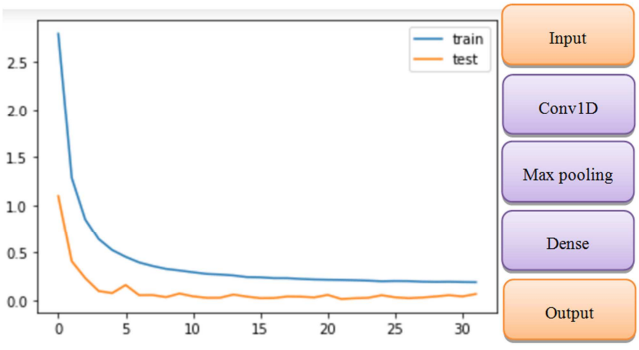

Figure 4. *Demonstration of the performance of the model over a period of time.*

After 63 epochs and have enough dataset with loss: 1.0385e-05 - accuracy: 1.0000 - val_loss: 4.9401e-06 - val_accuracy: 1.0000, performance of the model is not good.

Drawbacks of this method:

(i) Takes lots of time (time-consuming);

(ii) Contravention in some data (altered speaking manner) can ruin all of the datasets;

(iii) Cannot deal with environmental noise and distorted acoustics and speech correlated noise;

(iv) Low performance;

(v) Having weakness in changing the sample rate of a large number of the dataset;

This method is suitable to create a model for 10 to 50 or maybe more with enough dataset, but for a larger vocabulary, you need to follow another way. So that the drawbacks were taken into consideration and fueled further research which led us to a good ASR model.

3. Speech Recognition with CMU Sphinx

The dominant technological approaches for speech recognition systems are based on pattern matching of statistical representations of the acoustic speech signal, such as HMM whole word and subword (e.g., phoneme) models [10]. Statistical Language Modeling (LM) is the evolvement of probabilistic models that are qualified to predict the next word in the sequence according to the word before it, CMUSphinx uses an acoustic model, a dictionary, and an n-gram language model, which determines the phonetic units in the word available in the dictionary [11-15].

Speech recognition with CMUSphinx:

given the acoustic data: $X=x_1, x_2, x_3 \ldots. x_k$. Given the Word Sequence: $Wr=wr_1, wr_2, wr_3 \ldots. wr_k$. The target is to increase P(Wr/X). In agreement with Bayes' Theorem: P(Wr|X)=(P(X/Wr) P(Wr))/P(X)

Where:

P(X|Wr)=Acoustic model(HMMs)

P(Wr)=Language model.

P(X)=Constant for a complete sentence.

Sphinx2 uses dialog system language learning system and it is oriented on speech recognition in real time which makes it ideally suited for developing various mobile applications [16].

Sphinx3 represents semi continuous speech recognition acoustic model, adopted a common continuous model constructed on HMM [16]. Hidden Markov modeling of speech assumes that speech is a piecewise stationary process, that is, an utterance is modeled as a succession of discrete stationary states, with instantaneous transitions between these states [17].

CMUSphinx 4 is the latest addition which is differently designed from the earlier Sphinx systems regarding flexibility, modularity, and algorithmic aspects. You can modify the language model from a statistical N-gram language model to a context-free grammar (CFG) or a stochastic CFG by shifting only one portion of the system, meaning the linguist. Similarly, it is feasible to run the system using continuous, semi-continuous, or discrete phase output distributions by adequate rectification of the acoustic scorer. The overall architecture of sphinx 4 consists of the front-end, decoder, and knowledge base, which decoder itself consists of the search manager, the linguist, and the acoustic scorer. "Sphinx-4 puts out a beam pruner that limits the scores to a configurable least possible amount close to the best score, while also maintaining the total number of active tokens to a configurable maximum" [18].

Now let's move on to the work summary with CMUSphinx starting from the dataset, the database contains information that is required to extract statistics from the speech in form of the acoustic model. More than 10 hours of recorded words have been taken. Then filtered the dataset and prepared it for training. One of the factors that can affect the recognition process is the mismatch of the sample rate, all of your datasets must be 16 kHz (or 8 kHz, depending on the training data), you can use sox for this propose, "sox (Sound eXchange) is a cross-platform audio editing software. It has a command-line interface, and is written in standard C. It is free software, licensed under the GNU" [19]. Here it is momentous to prepare two dictionaries: "one in which legitimate words in the language are mapped to sequences of sound units (or sub-word units), and another one in which non-speech sounds are mapped to corresponding speech-like sound units" [20]. The file structure for the database is the following: you can name your folders and files whatever you want but be careful about extensions, I chose Sunshine.

Sunshine:

Sunshine.dic (Phonetic dictionary)

Sunshine.phone (Phone set file)

Sunshine.lm.DMP (Language model)

Sunshine.filler (List of fillers)

Sunshine_train.fileids (List of files for training)

Sunshine_train.transcription (Transcription for training)

Sunshine_test.fileids (List of files for testing)

Sunshine_test.transcription (Transcription for testing)

*.fileids:

The Sunshine_train.fileids and Sunshine_test.fileids files are text files that list the names of the recordings (utterance ids) one by one, do not include audio file extensions in their content [20].

*.transcription:

The Sunshine_train.transcription and Sunshine_test.transcription files are text files listing the transcription for each audio file:

One of the goals of this research paper is to guide you on how to create a phonetic dictionary for your own language. A phonetic dictionary provides the system with a mapping of vocabulary words to sequences of phonemes. To create a phonetic dictionary, you will come across ARPABET, ARPABET, or ARPAbet is developed by Advanced Research Projects Agency (ARPA) in the 1970s which is a collection of phonetic transcription codes. The purpose of developing ARPANET was to understand speech in the Research project. It demonstrated phonemes and allophones of General American English with preferable sequences of ASCII characters. So creating a phonetic dictionary requires knowledge about your own language phonology. In this paper the Dari language phonology is used and listed all of

the words and their pronunciation like زود "quick" /zuː d/ and زور "strength" /zoː r/, then I wrote down the words in English like دنیا - Dunya, after that I used Lexicon Tool which generates a pronunciation dictionary from a list of words in a form suitable for use with a speech recognizer, such as CMUSphinx. The Lexicon Tool uses the CMUdict dictionary along with some simple normalization and inflection rules to identify a word and uses letter-to-sound rules when all else fails. So here is the output of my work.

آب	AH B
ابتکار	EH B T AH K AH R
ابر	EY B ER
ابراز	EY B R AH Z
ابرو	AE B R OW
آبرو	AA B R OW
ابریشم	AE B R AH SH AH M
ابریشمی	AE B R EH SH AH M IY
ابزار	AE B Z AH R
ابعاد	AE B AE D
ابله	AE B L AH
آبله	AA B L AH
ابهام	EH B HH AH M
آبی	AE B IY
آبیاری	AH B AY AH R IY
آپارتمان	AH P AA R T AH M AH N
اتحاد	EH T AH HH AH D
آتش	AE T AH SH
اتفاق	EH T AH F AH K
اتمام	EH T M AH M
اتمسفیر	AE T M AH S F IH R
آچار	AH CH AA R
آخر	AE K AH R
آداب	AE D AH B
آدم	AE D AH M

From these words, a phoneme file has been created and then created a language model. The language model tells the decoder which sequences of words are possible to recognize. There are lots of tools like SRILM which is the most advanced toolkit up to date, CMUCLMTK, IRSLM, MITLM, web service such as Sphinx Knowledge Base Tool, and… but the problem is that some of them only support ASCII characters and English language and they do not support other languages, I personally prefer KenLM which can estimate, filters, and queries language models. Estimation is rapid and can be scaled on account of streaming algorithms. You can convert your model into binary format. After Setting up the training scripts and the format of database audio according to my needs, I started training in the Ubuntu environment and used Some additional scripts that will be launched if you choose to run them. These additional training steps can be costly in computation but improve the recognition rate. It's critical to test the quality of the trained database in order to select the best parameters, understand how your application performs, and optimize the performance. To do that, you need decoding. The Decoder takes a model, tests part of the database and reference transcriptions and, estimates the quality (WER) of the model. Within the testing phase, use the language model with the description of the possible order of words in the language. Here the result of running the decoder:

```
SENTENCE ERROR: 3.2% (264/8313)   WORD ERROR RATE: 3.3% (272/8313)
```

4. Speech Recognition with DeepSpeech

4.1. Reccurent Neural Network

Recurrent neural networks have been a significant hub of research and advancement since the 1990s. They are designed to indoctrinate ordinal or time-varying samples. A recurrent net is a neural network with feedback (closed-loop) connections [Fausett, 1994]. Examples include BAM, Hopfield, Boltzmann machine, and recurrent backpropagation nets [Hecht-Nielsen, 1990]. The architectures span from fully interconnected to partially connected nets, which include multilayer feedforward networks with different input and output layers. Learning is an essential feature of neural networks and a leading feature that creates a handy application using a neural approach, in addition, a Real-time determination for optimization problems is frequently necessary for scientific and engineering problems, including signal processing [21-24].

4.2. Deep Speech

DeepSpeech is an open-source voice recognition engine that is used to convert speech into text. It was using a recurrent neural network (RNN) to convert speech. To convert speech to text, a series of features must be extracted, so $x_{t,k}^{(n)}$ represents the power of the k^{th} frequency bin in the audio frame at time t. The main purpose of Recurrent Neural Network is to transform an input order "x" into a string of character probabilities for the transcription "y" [25]. The RNN model in DeepSpeech is consists of 5 layers of hidden units the first 3 layers are calculated by: $ht^{(l)}=g(W^{(l)}h^{(l-1)}t + b^{(l)})$, The fourth layer is a bi-directional recurrent layer which intends to apply a limit sequence to label each component of the sequence based on the element's past and future contexts [25]. The fifth layer is a non-recurrent that takes the forward and backward units, finally, the output layer is a standard softmax function that returns the predicted character probabilities for each portion of the time t and character "k" [25-28]. For computation, DeepSpeech uses CTC loss to measure the error in prediction. "Connectionist temporal classification (CTC) is a kind of neural network output that links scoring function, for training recurrent neural networks (RNNs) like LSTM networks so that the timing is variable and it holds sequence problems" [29].

4.3. Dataset Preparation and Training the Data

The dataset has been prepared, so 3 files needed to make ready the dataset: train.csv, dev.csv, test.csv. the CSV files included wav_filename, wav_filesize, and transcript, you can easily get the file size and audio file name using python. The following ratio for all audio files has been taken into account: 70 (training) – 20 (dev) – 10 (testing)! For training, you have to use Python 3.6, Deep-Speech, Tensorflow, and Mac or Linux environment.

To manage Python environments, it is good to create virtualenv. The wav files and their corresponding CSV files have been put into separate folders, then for language model creation again KenLM has been used and created a file filled

with audio files transcription and another file which feed up with pure Dari alphabet, a scorer file must be created which consists of 2 sub-members, a trie data and KenLM language model structure that contain all words in the vocabulary. Finally, the training has been started, after 33 epochs with a learning rate of 0.00095, train batch size 80, dev batch size 80, test batch size 40 and 375 number of hidden layers the output_graph.pb and output_graph.pbmm models have been created. Here is the result, for most of the words the WER is 1, loss less than 1, and CER less than 0.5.

5. Experimental Results

The major components and topics within the space of ASR are: 1) feature extraction; 2) acoustic modeling; 3) pronunciation modeling; 4) language modeling; and 5) hypothesis search [30].

To create the best possible models for a language and appraise the performances of the models, different experiments have been used by modifying the percentage of invisible audio files during the training. In this paper, you saw developing a speech recognition from scratch used CNN (conv1d) then went through the two most powerful open source frameworks which use different neural network algorithms with their pros and cons, for each application different number of the dataset have been used, for example with CNN 297482 sec audios, for CMUSphinx 10 hours recorded files have been used. The performance of the model with CMUSphinx and DeepSpeech speech was satisfactory.

In the first experiment, each neural network in cycles of 66 epochs has been trained, evaluating the resulting network after every cycle, but noticed a decrease in performance which can most likely be attributed to overfitting and underfitting. By lowering the number of the words to 50 40 30 20 10 5 3 and up to 2 the performance of the models got better. Then I got a subset of my own collected dataset and start training with CMUSphinx with the following configuration:

```
CFG_HMM_TYPE='.cont;
$CFG_FEATURE="s2_4x";
$CFG_NUM_STREAMS=4;
$CFG_INITIAL_NUM_DENSITIES=256;
$CFG_FINAL_NUM_DENSITIES=256;
$CFG_N_TIED_STATES=2000;
$CFG_MMIE="yes";
$CFG_G2P_MODEL='yes';
$DEC_CFG_VERBOSE=1
```

After training, I got two different folders with different files under the name of model architecture and model parameter following WER of 3.3% and noticed an increase in performance and got high accuracy, CMUSphinx is the best approach for speech recognition because with a fewer number of the dataset you can create a good model, also probabilistic works well as the problems with creating Speech-to-text model are the altered speaking manner, homophone, homograph and distorted acoustic like pray/prey. The drawback with CMUSphinx is creating a phonetic dictionary as ARPABET does not support other languages, you need to create one for your own language.

In addition, DeepSpeech has been used, models are trained for 33 epochs with a learning rate of 0.00095 on the full Dari dataset, it returned 1% WER that can be gratified for the created model. The drawbacks of DeepSpeech are: (i) require Linux or mac environment, (ii) usage of some old versions of the libraries, look at synopsis of studies in [31-36].

Ultimately, we can recap the factors that affect the recognition process: 1. Environmental noise, i.e., stationary or nonstationary additive noise; 2. Distorted acoustics and speech correlated noise; 3. Different microphones; 4. limited frequency bandwidth; 5. Altered speaking manner; 6. Homophone; 7. Homograph; 8. Phonetic dictionary; 9. Language model; 10. Dataset; 11. Sample rate; 12. Neural network algorithms; 13. The number of channels of the incoming audio; 14. language constraints, [37-41].

Table 1. *Presents the results of fine-tuning process.*

Models	WER	CER	loss	Learning rate
CNN (conv1d)	-	-	0.1867	0.0001
DeepSpeech	1	0.25	0.71	0.00095
CMUSphinx	3.3	3.2	-	-

6. Conclusion

According to the above, it can be concluded that neural network algorithms work well with ASR, I investigated the performance of an ASR based on CNNs, RNN, and HMM. This system was based on CMUSphinx from Carnegie Mellon University and DeepSpeech. DeepSpeech introduces an end-to-end deep learning-based speech system. Deepspeech is able to exceed in performance than existing state-of-the-art recognition pipelines and CMUSphinx has the flexibility in the usage of various kinds of acoustic and language representations, after that I created a phonetic dictionary for the Dari language, Finally, this project can conduce to subsequent studies and works on building the language model for different languages including Dari.

References

[1] https://www.nidcd.nih.gov/health/journey-of-sound-video.

[2] H. Satori, M. Harti, and N. Chenfour, Introduction to Arabic Speech Recognition Using CMUSphinx System.

[3] Mohamed Yassine El Amrani, M. M. Hafizur Rahman, Mohamed Ridza Wahiddin, Asadullah Shah, Building CMU Sphinx language model for the Holy Quran using simplified Arabic phonemes, Elsevier, 2016.

[4] Mohammed Dib, Automatic Speech Recognition of Arabic Phonemes with Neural Networks, Springer Nature Switzerland AG 2019.

[5] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu, Convolutional Neural Networks for Speech Recognition, IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, VOL. 22, NO. 10, OCTOBER 2014.

[6] O. Abdel-Hamid, L. Deng, and D. Yu, "Exploring convolutional neural network structures and optimization techniques for speech recognition," in Proc. Interspeech, 2013.

[7] T. N. Sainath, A. rahman Mohamed, B. Kingsbury, and B. Ramabhadran. Deep convolutional neural networks for LVCSR. In ICASSP, 2013.

[8] Iffat Zafar, Giounona Tzanidou, Richard Burton, Nimesh Patel, Leonardo Araujo, Hands-On Convolutional Neural Networks with TensorFlow, 2018.

[9] D. Nagajyothi, P. Siddaiah, Speech Recognition Using Convolutional Neural Networks, international Journal of Engineering & Technology, 2018.

[10] Herve´ Bourlard and Nelson Morgan, CONNECTIONIST SPEECH RECOGNITION A Hybrid Approach, KLUWER ACADEMIC PUBLISHERS, 1994.

[11] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling inspeech recognition. IEEE Signal Processing Magazine, 29 (November) 2012.

[12] T. Schultz and A. Waibel, "Language-independent and language-adaptive acoustic modeling for speech recognition," Speech Communication, vol. 35, August 2001.

[13] G. Dahl, D. Yu, L. Deng, and A. Acero, "Large vocabulary continuousspeech recognition with context-dependent DBN HMMs," inProc. IEEE Int. Conf. Acoust., Speech, Signal Process., 2011.

[14] R. G. Leonard and G. R. Doddington, "A database for speaker-independent digit recognition," in Proceedings of the International Conference on Acoustics, Speech and Sig nal Processing, vol. 3. IEEE, 1984.

[15] Abhishek Dhankar, Study of deep learning and CMU sphinx in automatic speech recognition, IEEE, 2017.

[16] Rami Matarneh, Svitlana Maksymova, Vyacheslav V. Lyashenko, Nataliya V. Belova Speech Recognition Systems: A Comparative Review, IOSR Journal of Computer Engineering (IOSR-JCE), Volume 19, Issue 5, Ver. IV (Sep.-Oct. 2017).

[17] Renals, S., Morgan, N., Bourlard, H., Cohen, M., and Franco, H. Connectionist probability estimators in HMM speech recogni tion. IEEE Transactions on Speech and Audio Processing, 1994.

[18] Paul Lamere, Philip Kwok, Evandro B. Gouvea, Bhiksha Raj, Rita Singh, William Walker, Peter Wolf, The CMU Sphinx-4 Speech Recognition System.

[19] https://en.wikipedia.org/wiki/SoX.

[20] https://cmusphinx.github.io/wiki/.

[21] Andrew W. Senior and Anthony J. Robinson, "Forward backward retraining of recurrent neural networks," in NIPS, 1995.

[22] Alex Graves, Abdel-rahman Mohamed and Geoffrey Hinton, SPEECH RECOGNITION WITH DEEP RECURRENT NEURAL NETWORKS, [cs. NE] 22 Mar 2013.

[23] M. Schuster and K. K. Paliwal, "Bidirectional Recurrent Neural Networks," IEEE Transactions on Signal Processing, vol. 45, 1997.

[24] Yasser Mohseni Behbahani*, Bagher Babaali, and MussaTurdalyuly, Persian sentences to phoneme sequences conversion based on recurrent neural networks, Open Comput. Sci. 2016.

[25] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, Jie Chen, Jingdong Chen, Zhijie Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Ke Ding, Niandong Du, Erich Elsen, Jesse Engel, Weiwei Fang, Linxi Fan, Christopher Fougner, Liang Gao, Caixia Gong, Awni Hannun, Tony Han, Lappi Vaino Johannes, Bing Jiang, Cai Ju, Billy Jun, Patrick LeGresley, Libby Lin, Junjie Liu, Yang Liu, Weigao Li, Xiangang Li, Dongpeng Ma, Sharan Narang, Andrew Ng, Sherjil Ozair, Yiping Peng, Ryan Prenger, Sheng Qian, Zongfeng Quan, Jonathan Raiman, Vinay Rao, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Kavya Srinet, Anuroop Sriram, Haiyuan Tang, Liliang Tang, Chong Wang, Jidong Wang, Kaifu Wang, Yi Wang, Zhijian Wang, Zhiqian Wang, Shuang Wu, Likai Wei, Bo Xiao, Wen Xie, Yan Xie, Dani Yogatama, Bin Yuan, Jun Zhan, Zhenyao Zhu, Deep Speech 2: End-to-End Speech Recognition in English and Mandarin, Baidu Silicon Valley AI Lab1, 1195 Bordeaux Avenue, Sunnyvale CA 94086 USA Baidu Speech Technology Group, No. 10 Xibeiwang East Street, Ke Ji Yuan, Haidian District, Beijing 100193 CHINA.

[26] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, Andrew Y. Ng, Deep Speech: Scaling up end-to-end speech recognition, arXiv: 1412.5567v2 [cs. CL] 19 Dec 2014.

[27] Willem R¨opke, Roxana R˘adulescu, Kyriakos Efthymiadis, and Ann Now´e, Training a Speech-to-Text Model for Dutch on the Corpus Gesproken Nederlands.

[28] https://deepspeech.readthedocs.io/en/r0.9/.

[29] https://en.wikipedia.org/wiki/Connectionist_temporal_classification.

[30] L. Deng and X. Li, "Machine learning paradigms for speech recognition: An overview," IEEE Trans. Audio, Speech, Lang. Process., vol. 21, no. 5, May 2013.

[31] Naveen Srinivasamurthy and Shrikanth Narayanan, Language-Adaptive Persian Speech Recognition, 2003.

[32] Sun R., Giles C. L., Sequence learning: from recognition and prediction to sequential decision making, IEEE Intelligent Systems, 2001.

[33] L. R. Medsker, Departments of Physics and Computer Science and Information Systems American University, Washington, D. C., L. C. Jain, Knowledge-Based Intelligent Engineering Systems Centre, Faculty of Information Technology, Director/Founder, KES, University of South Australia, Adelaide, The Mawson Lakes, SA, Australia, Recurrent neural network, crs 2001.

[34] Demuynck, K., Roelens, J., Compernolle, D. V., Wambacq, P.: Spraak: An open source" speech recognition and automatic annotation kit". In: Ninth Annual Conference of the International Speech Communication Association (2008).

[35] A. J. Robinson, "An Application of Recurrent Nets to Phone Probability Estimation," IEEE Transactions on Neural Networks, vol. 5, no. 2, 1994.

[36] Ben Shneiderman, the Limits of Speech Recognition, COMMUNICATIONS OF THE ACM September 2000/Vol. 43, No. 9.

[37] Dr. Raj Reddy, spoking language processing, 2001.

[38] Robert V. Shannon; Fan-Gang Zeng; Vivek Kamath; John Wygonski; Michael Ekelid, Speech Recognition with Primarily Temporal Cues, Science, New Series, Vol. 270, No. 5234. (Oct. 13, 1995).

[39] FRANÇOIS CHOLLET, Deep Learning with Python, Manning Publications Co.

[40] B. Logan and A. Salomon, "A music similarity function based on signal analysis," in ICME 2001, August 2001.

[41] Nitin Indurkhya (Editor), Fred J. Damerau (Editor), Handbook of Natural Language Processing, Second Edition, Chapman & Hall/CRC, Machine Learning and Pattern Recognition Aeries.

[42] The credit for the first figure goes to Rolphe Frédéric Fehlmann, it is designed by Dunya Yousufzai but the concept was taken from Rolphe Frédéric Fehlmann.

Biography

Dunya Yousufzai is a software engineer. Prior to her recent paper, she has written a book for women empowerment. In the 10th Afghanistan national science project competition which was held on June 2015 and organized by the ministry education of Islamic republic of Afghanistan and ATCE (NGO), she obtained a golden medal. Amity Institute for competition examination certifies that she participated in the 3rd global talent search examination held on 23rd November 2014 from India. Global Hippo Association in Italy certifies that she successfully participated in Hippo's 3rd international language competition. Pedagogical Association education without border in Bulgaria certifies that she was one of the successful participants who obtained a gold medal in one of the international competitions.