Parasitic sorority of speech processing algorithms with an assortment of statistical toolkits

Prathibha Sudhakaran¹, Ashwani Kumar Yadav², Sunil Karamchandani ³

¹Research Scholar, ASET, Amity University Jaipur, Rajasthan, India
²Assistant Professor, ASET, Amity University, Jaipur, Rajasthan, India
³Associate Professor, DJ Sanghvi College of Engineering, Mumbai, Maharashtra, India
¹prathibharsamity@gmail.com, ²ashwaniy2@gmail.com, ³skaramchandani@rediffmail.com

Abstract. Speech is a one-dimensional quasi non-stationary time varying signal produced by a sequence of sounds. Speech signals are random in nature. Speech signals are easily corrupted by noise so recognition is an important role in speech processing. Many researches have designed recognition system with challenging parameters. Speech corpus can vary from environment, region, dialects, age, rate at which words are spoken. Pre-processing is the first step which includes framing, de-noising and filtering. This paper focuses on speech techniques and statistical open source tools such as HTK, Julius, CMUSphinx and Kaldi. The word error rate obtained using all the toolkits on WSJ1 corpus gives us a clear understanding that Kaldi stands out as the most advanced recipes and scripts for speech recognition systems. An Indian English corpus by IITM was implemented in Kaldi yeilds WER of 6.41 and has been compared to other indian and international languages and well known corpuses.

1. Introduction

Automatic Speech Processing is an important part of information and communication technology today. According to the development in technology most of the highly efficient companies use robots and human machine interacting systems. Applications include biometric, security, mobile, healthcare, video games, weather forecasting, transcription etc. It has been an interesting and profound topic over decades.[¹] It is indeed a great challenge to make a computer to understand spoken language since a word cannot be repeatedly spoken by the same speaker with same slang, pitch and parameters. Vowels, semi-vowels, nasal consonants, unvoiced fricatives, voiced fricatives, voiced and unvoiced stops, diphthongs are the diverse sounds in speech. Every language has its own set of alphabets, vowels and consonant. Figure 1 shows the different types of speech recognition systems.[²]
The main parts of the ASR system include the front end processing, acoustic model, lexicon, language model and the decoder as shown in Figure 2.[3] The front end is involved in pre-processing and feature extraction. Decoder has the property to search through all the word sequences that is expected to generate. They are estimated by the acoustic model. Currently research is focused on continuous speech for many languages with the advent of machine learning and artificial intelligence. Acoustic modeling is the heart of any speech system. Speech recognition systems use Bayes theorem for continuous speech recognition as the main problem is to find the word sequence: the equation of which is given by the equation below:

\[ W = \arg \max \left( P(W|O) = \arg \max \left( P(W)P(O|W)/P(O) \right) \right) \]

where, \( P(W) \) = prior probability of word \( W \)
\( P(O|W) \) is the acoustic model and
\( P(O) \) is the probability of acoustic observation.
2. Open Source Toolkits

A growing demand for open source toolkits is emerging. In this section a comparison between the popular toolkits available. Kaldi outperforms all the other recognition toolkits. The Sphinx toolkit also provides a training pipeline, not as advanced as the Kaldi pipeline, but with the possibility to generate good results shortly. HTK comes with the simplest start-up kit, the buildup of a self-trained recognition system to reach the performance. Sphinx requires the development of advanced parts by the user and extensive tuning.

Table.1 Comparison between the various toolkits based on the programming language, operating systems and the vocabulary.

| Toolkit        | Programming Language | Operating System    | Large Vocabulary |
|----------------|----------------------|---------------------|------------------|
| CMUSphinx      | C                    | Open Source         | Small/Medium     |
| Sphinx4        | Java                 | Open Source         | Yes              |
| SphinxTrain    | C                    | Open Source         | Yes              |
| HTK            | C                    | Open Source         | Yes              |
| Julius         | C, Python            | Open Source         | Yes              |
| Kaldi          | C++, Python          | Open Source         | Yes              |

2.1 CMU Sphinx Toolkit

CMUSphinx developed at Carnegie Mellon University has a speech recognition system written in C and Java. The code is available at SourceForge. It is highly portable. It has been found suitable to be applied to isolated or connected digits and also for small and medium vocabulary. [4] Hasan Satori et.al used CMUSphinx for Arabic Speech Corpus. The corpus consists of different individuals who utter 10 Arabic digits. The average accuracy is around 96.67%. This projects the adaptability of the CMU Sphinx to the Arabic language. [5] Hussein Hayassat et.al have used the CMUSphinx toolkit for Arabic language. The corpus consists of 2 lakh unique words and attains a WER of 0.787% and an accuracy of 99.213%. The size of the speech corpus is an important factor which determines the accuracy of any ASR system. [6] Mohammad A M Abushariah also used CMUSphinx toolkit for Arabic language. The corpus consists of 8043 utterances and has attained an average accuracy of 92.67%. More the data, more will be the accuracy of the model. [7]
2.2 HTK Toolkit
Its based on Balm Welsch algorithm. It uses HMM (Hidden Markov Model) for training testing and for result analysis.

Mohit Dua et.al have incorporated the HMM based HTK toolkit to the Punjabi corpus. Stephen J Young et.al have used a credit card corpus containing spontaneous speech using the HMM based HTK toolkit. The overall results obtained very not satisfactory but it did take into account laughs, fast speech and lazy speech effects. [8] Shweta Tripathy et al. modeled a Hindi speech recognition system with HTK. The training data of 1225 and testing data of 175 utterances formed the corpus. The accuracy achieved was 88% and 90% with different speakers.[9]

Ananthakrishna T et.al developed a Kannada speech recognition system using HMM based HTK toolkit. MFCC are computed in the acoustic processing. The size of the dictionary has 110 words. Baum Welch algorithm was used to train HMM and Viterbi algorithm for the decoding process. 97.1% of the word accuracy for syllable level and 98.6% for phone level were obtained. It was thus concluded that the syllable level HMMs perform better as compared to the phone level.[10]

2.3 Kaldi Toolkit
The goal of this open source toolkit is to have a modern and flexible code that is easy to understand modify and extend. Kaldi uses DNN (Deep Neural Network) for acoustic modeling [11]. Kaldi works on the Viterbi Algorithm and offers scripts and recipes to researchers in speech technology. Tools such as arpa2fst, fstcompile, and multiple Perl script are used in Kaldi. It is always updated with recent techniques like deep neural networks and bottleneck features (BNF).

Ahmed Ali et.al developed an Arabic speech recognition system using the Kaldi toolkit. The corpus consists of 194 hours of data (training set) and 9 hours (testing set). The WER recorded was 15.81% [12]

Biswajit Karan et.al have implemented an ASR system using Odia Language. The Corpus consists of 2130 utterances. WER obtained was 1.1. The number of utterances is less for this system. The proposed model can be used for LVCSR (Large Vocabulary Continuous Speech Recognition) [13]

MircoRavanelli et.al draws our attention to the interface between PyTorch and Kaldi. Pytorch is used to build neural networks in python language. It is capable of achieving good results over a variety of datasets. [14]

The feature extraction techniques incorporated in Kaldi is using MFCC and PLP. The relation between frequency and Mel scale is:

$$Mel(f) = \frac{2595 \log_{10}[1+f/700]}{f}$$

They use the Mel-scale filter banks. PLP on the other hand removes the irrelevant parts improving the recognition rate. The power spectrum in PLP for a windowed signal is given by:

$$P(w) = \Re(S(w)^2) + \Im(S(w)^2)$$

frequency warping into the bark scale is applied. The table 2 shows the comparison between the various languages the toolkits have been implemented and their acquired WER.

Table 2: Overview of the toolkits implemented in various languages with their WER and tokens

| Language  | Toolkit  | Token/Corpus | WER(W)/Accuracy(A)/Recognition Rate (R) |
|-----------|----------|--------------|----------------------------------------|
| Odia      | Kaldi    | 147 words    | 1.1 (W)                                |
| Kannada   | Kaldi    | 100 words    | 4.27 (W)                               |
| Kannada   | HTK      | 110 words    | 98.6% (A)                              |
| Arabic    | CMUSphinx| 500 voices   | 96.67% (R)                             |
| Japanese  | Julius   | 20K words    | 92.8% (A)                              |
3. Outcomes from the Literature

Kaldi which appeared in 2011 implements various features and scripts. It has features like deep neural models which provide the state of the art accuracy compared to the other toolkits. HTK toolkit requires and demands more effort and knowledge. Adaptation and Discriminative training are possible but tool chain is impossible without being an actual expert in the domain. Kaldi stands out as an outstanding performer in comparison to others and has set up a revolution in Open source ASR systems. The training tools in Sphinx though similar to Kaldi lacks in accuracy but enables training after installation. [3]

Fatima Barkani et.al have explained how ASR systems have evolved over time and they have been categorized from generation 1 to Generation 5. Initially the ASR systems were restricted to isolated words or were implemented for recognizing sounds using some ad-hoc methods. The acoustic – phonetic approaches led to recognizing of phonemes and digits [15]. Linear predictive coding and pattern recognition led to experiments on connected word sequences. The use of HMM models, neural networks further enhanced the performances of ASR systems. The study of the various open source toolkits based on HMM show their ability to many larger vocabularies and the improvements in performances of WER. The performance parameters would be the size or nature of the corpus/database, conditions of the data recorded, type of feature extraction used for training and the algorithm implemented in the acoustic, language models and in the design of the decoder.[16]

WER is the measuring tool in ASR systems. It is given by

\[
\text{WER} = \frac{(D+S+I)}{N} \times 100\% \tag{3}
\]

Where \(N\) is the number of the words used in the test
\(D = \) Number of deletions
\(S = \) Number of Substitutions
\(I = \) Number of Insertions

A study of many tools used in ASR concludes that Kaldi has emerged to be the toolkits which can give us optimum results even over large vocabulary. The acoustic modeling in Kaldi involves the generation of monophone, triphone, delta, LDA and MLLT estimations. Kaldi enable N-gram modeling efficiently for a given language model.

Christian Gaida et.al have concluded that the WER after having applied to the WSJ1 Corpus obtained the following results as shown in figure shown below. Kaldi shows the least WER and is the most significant and advanced techniques which have been added as a contribution to their paper. Sphinx though not as advanced as Kaldi but also provides good results. HTK has a self-trained startup kit unlike Sphinx. [16]

![Figure 3. Graphical representation indicating the WER performance of the various Speech Toolkits](image-url)
4. Implementation of Kaldi for Indian English

In the experiment Kaldi toolkit is used on an Indian English speech database which consists of 15459 lines and 9377 unique words for monophone and triphone models and compared the accuracy to other developed language systems. Kaldi is based on Viterbi and Clustering algorithm. The speech database consists of men and women of different linguistic background reading sentences. The lexicon file was created using the ILSL set (India language speech sound label set). Kaldi uses HMM in conjunction with GMM or DNN. We can implement n-gram model in the scripts in Kaldi allow us to train 3 types of context dependent phones denoted as tri1, tri2 and tri3. The system did not model any non-speech events. Non-speech events include background noise, vocal noise or 'hmm’ sound by the speakers. Hence it can be concluded that the model created ignored the non-speech sounds. MFCC was used to represent the acoustic characteristics of a frame of speech. MFCC was evaluated using:

\[ Mel(f) = 2595 \log_{10} \left( 1 + \frac{f}{700} \right) \]  

In tri1 (\( \Delta + \Delta \Delta \)) first the differentiation of the signal features known as \( \Delta \) which is given by:

\[ \Delta_k = f_k - f_{(k-1)} \]  

Second differentiation is given by:

\[ \Delta \Delta_k = \Delta_k - \Delta_{k-1} \]  

In tri2 (LDA + MLLT) is used. Linear Discriminant Analysis (LDA) is a dimensionality reduction and classification approach. Based on maximizing of the ratio of inter class variance to within class variance or maximization of the ratio of overall variance to within class variance, LDA can be class-dependent or class-independent. [24]

In tri3 (LDA+MLLT+SAT) is implemented. SAT(Speech adaptive Training) for n-gram language model has been observed to further enhance the accuracy. [23]

Based on the study a 7-fold experimentation was performed which gives a WER of 6.41 corresponding to tri3 model. The performance has considerably shown improvement from monogram to trigram model. The following table 3 shows the WER of the system implemented:

|  | Mono | Tri1 \((\Delta + \Delta \Delta)\) | Tri2 \((\text{LDA} + \text{MLLT})\) | Tri3 \((\text{LDA} + \text{MLLT} + \text{SAT})\) |
|---|---|---|---|---|
|  | 16.97 | 7.98 | 7.61 | 6.41 |

5. Comparison of Kaldi for different languages

L.Babu et.al has chosen MFCC features for the monophone model. 1200 sentences were selected from 5114 sentences of a story for processing from 18 speakers, 9 male and 9 female. The testing however was done with AIR read data. The IRSTLM toolkit was used to build the language model. The LDA+MLLT triphone 3 model shows better performance compared to the \( \Delta + \Delta \Delta \) model triphone l with a WER of 34.45 [17] P. Upadhyay et.al has given a comparative analysis on the various speech corpuses using Kaldi toolkit. The size of these corpus may vary but each has used MFCC as the baseline for comparison of Italian, arabic, english and Hindi. The phone to audio alignments of TIMIT database was 39 english phones, it was 54 for hindi phones. Lesser the number of phone models resulted in better performance. [18] Similar experiments have been conducted for Kannada language and shows a WER of 12.72 for one
data-set and 4.54 for another data set. [19] For punjabi language using kaldi MFCC and PLP are used for feature extraction using 2-gram and 3-gram LM models. For tri3 (LDA+MLLT+SAT) the WER is 21.2 for MFCC and 22.7 when PLP features were used. [20,21] Kaldi toolkit implemented for a 3 fold validation experiments in Marathi language gives an average WER of 21.2 for sGMM (subspace Gaussian mixture model). Triphone model the WER was 24.2, considerable improvement has been recorded. [22]

| Language  | Corpus                                       | WER [ tri3] |
|-----------|----------------------------------------------|-------------|
| Malayalam | 1200 sentences , 9 female and 9 male speakers| 34.45       |
| Marathi [21] | Text corpus-340 sentences, 50 speakers, 34 districts | 24.2 |
| Arabic [18] | Gale corpus                                 | 22.32       |
| Punjabi [20] | 1000 sentences                             | 21.2        |
| English [18] | TIMIT                                      | 18.4        |
| Kannada [19] | 110 words in short sentences                | 12.72       |
| Hindi [18] | AMUAV                                       | 12.62       |
| Italian[18] | ChilIt corpus                              | 8.6         |

6. Conclusion

An ASR system converts speech into text with the help of an algorithm or computer program. This paper gives an overview of the various recognition techniques implemented successfully by researchers in many unique databases and the evolution of changes in the area. In sections 2 the focused is on the widely used open source speech recognition toolkits and compared their performances. Kaldi is observed to have better WFST and math support compared to the other toolkits. Section 3 discusses about the findings of the literature survey done on the open source toolkits. In Section 4 Kaldi is implemented to an Indian English corpus from IITM speech lab resulting into a WER of 6.41. The recognition accuracy of a triphone model based on words before and after a phoneme is much better that monophone which is based on a single sound
or phoneme. One can improve the WER and get more robust results by using a large corpus and implementing DNN. The performance of a speech system is challenging as it depends on the variations in speakers, their pronunciations, the rate at which they speak and the dialects of the regions they belong to. Hence with the advancements in open source toolkits and their fast processing speed a considerable change in the overall accuracy in the speech models is observed. We hereby conclude that the FST framework and the recipes thus enable Kaldi to adjust to all relevant corpsuses.

[1] S. S., “Performance of Different Classifiers in Speech Recognition,” Int. J. Res. Eng. Technol., vol. 02, no. 04, pp. 590–597, 2013, doi: 10.15623/ijret.2013.020432.
[2] L. R. Rabner and R. W. Schafer, “Introduction to digital speech processing,” Found. Trends Signal Process., vol. 1, no. 1–2, pp. 1–194, 2007, doi: 10.1561/2000000001.
[3] C. Gaida et al., “Comparing Open-Source Speech Recognition Toolkits,” Tech. Rep., DHBW Stuttgart, 2014, [Online]. Available: http://www.suendermann-oefl.com/wp/pdf/oasis2014.pdf.
[4] R. Singh, W. Walker, and P. Wolf, “The CMU Sphinx-4 Speech Recognition System,” Evaluation, pp 2-5
[5] H. Satori, H. Hiyassat, M. Harti, and N. Chenfour, “Investigation Arabic speech recognition using CMU Sphinx system,” Int. Arab J. Inf. Technol., vol. 6, no. 2, pp. 186–190, 2009.
[6] H. Hyassat and R. Abu Zitar, “Arabic speech recognition using SPHINX engine,” Int. J. Speech Technol., vol. 9, no. 3–4, pp. 133–150, 2006, doi: 10.1007/s10272-008-9009-1.
[7] M. A. M. Abushariah, R. N. Ainon, R. Zainuddin, M. Elshafei, and O. O. Khalifa, “Native speaker-independent Arabic speech recognition system based on Hidden Markov Models using Sphinx tools,” Int. Conf. Comput. Commun. Eng. ICCCE’10, no. May, pp. 11–13, 2010, doi: 10.1109/ICCCE.2010.5556829.
[8] S. J. Young, P. C. Woodland, and W. J. Byrne, “Spontaneous Speech Recognition for the Credit Card Corpus Using the HTK Toolkit,” IEEE Trans. Speech Audio Process., vol. 2, no. 4, pp. 615–621, 1994, doi: 10.1109/89.326619.
[9] S. Tripathy, N. Baranwal, and G. C. Nandi, “A MFCC based Hindi speech recognition technique using HTK Toolkit,” 2013 IEEE 2nd Int. Conf. Image Inf. Process. IEEE ICIP 2013, pp. 539–544, 2013, doi: 10.1109/ICIP.2013.6707650.
[10] “Kannada Word Recognition System Using HTK,” pp. 1–5, 2015.
[11] J. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P. and Silovskysky, “The Kaldi speech recognition toolkit (No. CONF),” IEEE Signal Process. Soc., pp. 1–4, 2011, [Online]. Available: http://kaldi.sf.net/.
[12] A. Ali, Y. Zhang, P. Cardinal, N. Dahak, S. Vogel, and J. Glass, “A complete KALDI recipe for building Arabic speech recognition systems,” 2014 IEEE Work. Spok. Lang. Technol. SLT 2014 - Proc., pp. 525–529, 2014, doi: 10.1109/SLT.2014.7087629.
[13] B. Karan, J. Sahoo, and P. K. Sahu, “Automatic speech recognition based Odia system,” 2015 Int. Conf. Microwave, Opt. Commun. Eng. ICOICE 2015, pp. 353–356, 2016, doi: 10.1109/ICOICE.2015.7489765.
[14] M. Ravanelli, T. Parcollet, and Y. Bengio, “The Pytorch-kaldi Speech Recognition Toolkit,” ICASSP. IEEE Int. Conf. Acoust. Speech Signal Process. - Proc., vol. 2019-May, pp. 6465–6469, 2019, doi: 10.1109/ICASSP.2019.8683713.
[15] V. Ramesh, V. C. Jaunky, R. Roopchund, and H. S. Oodit, ‘Customer Satisfaction’, Loyalty and ‘Adoption’ of E-Banking Technology in Mauritius, vol. 1076, 2020.
[16] P. K. Sahu and D. S. Ganesh, “A study on automatic speech recognition toolkits,” 2015 Int. Conf. Microwave, Opt. Commun. Eng. ICOICE 2015, pp. 365–368, 2016, doi: 10.1109/ICOICE.2015.7489768.
[17] L. B. Babu, K. R. Sreelakshmi, A. George, and L. Mary, “Continuous speech recognition system for malayalam language using Kaldi,” 2018 Int. Conf. Emerg. Trends Innov. Eng. Technol. Res. ICETETR 2018, pp. 1–4, 2018, doi: 10.1109/ICETETR.2018.8529045.
[18] P. Umapathy, O. Farooq, and Y. Vardhan, “Continuous Hindi Speech Recognition Using Kaldi ASR based on Deep Neural Network I am currently working on Audio Visual Speech Recognition in Hindi Language View project Currently working on Speller BCI for Hindi (Indian Language) View project,” no.
February, 2018, doi: 10.13140/RG.2.2.16897.97126.

[19] T. Sahana, N. Srilasya, K. Jeeva Priya, D. Gupta, and S. Vinay, “Comparison of different Acoustic Models for Kannada language using Kaldi Toolkit,” 2018 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2018, pp. 2415–2420, 2018, doi: 10.1109/ICACCI.2018.8554586.

[20] Mayank Singh and Viranjay M. Srivastava, “Implementing Architecture of Fog Computing for Healthcare Systems based on IoT,” International Journal of Engineering and Advanced Technology (IJEAT), vol. 8, no. 4c, pp. 23-27, April 2019.

[21] J. Guglani and A. N. Mishra, “Continuous Punjabi speech recognition model based on Kaldi ASR toolkit,” Int. J. Speech Technol., vol. 21, no. 2, pp. 211–216, 2018, doi: 10.1007/s10772-018-9497-6.

[22] S. Paulose, S. Nath, and K. Samudravijaya, “Marathi Speech Recognition,” in SLTU, 2018, pp. 235–238.

[23] Upadhyaya, Prashant, et al. “Speaker adaptive model for hindi speech using Kaldi speech recognition toolkit.” 2017 International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT). IEEE, 2017.

[24] Ali, Hazrat, et al. "Speaker recognition with hybrid features from a deep belief network." Neural Computing and Applications 29.6 (2018): 13-19.