Automatic Hindi Speech Recognition: Challenges and Future Scope

Ankit kumar, Subhash Chandra Gupta, Shashwat Tripathi, Vanshika Lamba, Aditya Singh

KIET Group of Institutions, Delhi-NCR, Ghaziabad, India

Corresponding author: Subhash Chandra Gupta, Email: subhash_g99@yahoo.com

Automatic Speech Recognition (ASR) field is the witness of remarkable improvement over the last few years. Deep learning architectures are the main reason for ASR evaluation. Deep learning techniques tremendously improve ASR performance, but these architectures are data-hungry. There are more than 5000 languages in India that do not have such a huge amount of resources. Hindi is one of them. The lack of freely available Large Vocabulary Hindi speech datasets is the major hurdle for developing Hindi ASR. In this paper, we investigate the various work available for Hindi ASR. The available resource is also discussed with issues and challenges in Hindi ASR. This paper provides a detailed analysis of various techniques available in ASR with comparison to Hindi ASR. The future scope of Hindi ASR is also presented.

Keywords: Perpetual, Recognition, Neural, Spectral, Convolution.

2023. In Saroj Hiranwal & Garima Mathur (eds.), *Artificial Intelligence and Communication Technologies*, 1113–1122. Computing & Intelligent Systems, SCRS, India. https://doi.org/10.52458/978-81-955020-5-9-105

1. Introduction

Automatic Speech Recognition (ASR) is the process of taking speech utterance and converting it into text sequence as close as possible. There are many functional areas in ASR; some are as follows: dictation, a program control application, dialog systems, audio indexing, speech-to-speech translation, and query-based information retrieval system i.e., weather information system or some travel information system. With the increase in need of end-user focused applications such as look for voice and voice communication with the cellular device and domicile amusement systems, the robust speech recognition that works in all the real-world noises and other acoustic distorting conditions is in demand [1]. In our country, India, we have language variation within a few kilometers, and that is why we have 22 important languages with around 750 dialects. To have a speech recognition system that can help remove the language barrier between Indian people is really a good thing that can happen to them. The speech recognition system's focus is to help the system identify the voice signal and transform the spoken word from voice to textual form. Google has created Google Assistant, who can converse in 120+ global languages consisting of Indian regional languages like Tamil, Telugu, Marathi, Malayalam, Kannada, and Hindi. The ASR engine built by Google or Apple trained over thousands of hours of speech data. Only a few languages like English, Chinese have such amount of huge data.

In India, the majority of languages do not have such resources. Only a handful of the resource is available for research. The Hindi language also does not have a freely available large vocabulary speech dataset. Few hours of speech data is available by TIFR, Mumbai [ref], and TDIL [footnote], India, for Hindi speech recognition. Except for the speech resource, there are many other challenges for developing Hindi ASR. To implement an ASR system, some difficulties arise due to variations in speaking style and noise in the environment. So, the core objective behind developing an automatic speech recognition system is to convert a speech utterance into text sequence free of the device, independent of the speaker or the surrounding environment efficiently and accurately. For an efficient ASR system, the issues of speech robustness must be carefully studied and resolve. With the advancement of technology, ASR has become a more challenging area [2]. In ASR, a rigorous training procedure is followed to map the basic unit of speech to the acoustic observation. But the main factor for underlying degradation of system performance is the presence of additive or convolutive noise arising from the environment, channel interference, or encoding decoding process. Many approaches to maximize the ASR accuracy have been developed, but still, there is a huge gap in performance when the acoustic environment changes. Hindi speech recognition is not extensively examined, and very few works have been done in this area. In this work, we give a detailed review on Hindi speech recognition with their challenges and future scope of this field. This article would be a great starting point for those who are willing to work in Hindi speech recognition.

The rest of the paper is organized as follows: Section 2 describes the challenges and issues of Hindi speech recognition. Section 3 presents the State-of-the-art (SOTA) acoustic modeling techniques. Section 4 describes the available speech corpus for Indian languages. Section 5 compares the previous work on Hindi speech recognition. Section 6 reports the conclusion and future work.

2. Literature Review & Previous Work Comparison

We studied various state of art approaches proposed for robust speech recognition and continuous Hindi speech recognition system. We did comparison based on feature extraction, language modeling and acoustic modeling techniques. The standard feature extraction technique is Mel Frequency Cepstral Coefficient (MFCC) proposed by Davis & Mermelstein in 1980. Except MFCC, Perceptual Linear Prediction (PLP) and the MF-PLP are the most suitable choice for feature extraction. Other potential method which can be used for parameterization is based on wavelet transform. The wavelet packet transform was first employed by Erzin et al. [3] in 1995 for the computation of the spectrum in the speech recognition area. Several variants of MFCC and PLP were proposed in literature. For instance, Gammatone filters [4] and Gabor filters [5] were proposed as an alternative option for auditory speech analysis to derive optimal time-frequency resolution. Noise resistant features techniques take a focus on the effect of noise rather than on the removal of noise. Noise resistant

features can be find by neural network approaches such as artificial neural network- Hidden Markov Model (ANN-HMM) hybrid system [6], TANDEM [7], Bottleneck features [8] and by auditory based features techniques. These techniques usually produce better results than MFCC because these features have much more complicated generation process. Relative spectral processing (RASTA) use same idea and give better results. RASTA is also combined with PLP resulting RASTA-PLP for improvement in speech recognition. Till now there is no ubiquitous auditory theory about which kind of auditory based information is suits well for the robust speech recognition [1]. This makes the scope in finding the best auditory features for speech recognition. In the last few years, it has been observed that the use of ANN for finding effective features has increased tremendously [6, 7, 8] and gain satisfactory improvement in performance. Due to the capability of ANN modelling power in small regions of features space and non- linear modelling power which might be able to normalize data from different sources well [1, 7]. A 5- layer Multi-layer perceptron which has very narrow middle layer (bottle-neck) is usually used to extract the bottle-neck features [8] and these features perform better than TANDEM features. Even though existing investigation of BN features is not focused on noise robustness, therefore, huge possibilities that BN features are able to execute well on noise-robust speech recognition [1]. In more recently, CD- DNN-HMM [9] is different from TANDEM and help to achieving the huge accuracy improvement [9]. Up to now, in front- end or feature domain we concise our research direction in future to BN features[8] and CD-DNN-HMM [7] for finding noise resistant features for robust speech recognition. Another way to deal with noise is compensation using prior knowledge about distortion. There are many techniques which are used to learn the mapping between noisy speech and clean speech, some used stereo data e.g. Empirical Cepstral Compensation [10], Stereo-based Piecewise Linear Compensation for environment (SPLICE) [11] and some use multi-environment data e.g. Linear model combination [12, 13], source separation [14, 15, 16]. A recurrent neural network (RNN) is also use to extract the clean speech from noisy speech by modeling temporal signal [17]. With this quality of non-linear modeling power, RNN proved its effectiveness in noise cleaning process. RNN is also improved with a bidirectional long term memory (BLSTM) structure [18] which leads toward the efficient noise to clean speech mapping. A simple method to deal with noise in acoustic environment is multi-style training, which uses the noisy speech data to train the acoustic model. And the idea behind this is that any one of the noise types which we will use in training scenario will appear in the testing scenario. There are several major problems with multi-style training and these problems can be deal with the linear model combination or source separation techniques. In source separation is a source division approach and non-negative matrix factorization (NMF) has been exposed to be a very flourishing scheme [19] which is helpful in production of noise robust ASR. Although the traditional NMF-based approaches [20, 21] can be used to get state-of–art performance, other constrained can be imposed into NMF to obtain more gain [22].

Acoustic Modeling plays a vital role in speech recognition systems. The statistical methods like HMM and GMM are the most popular choice for developing ASR system [23]. Huge research has been done to increase the evaluation speed of GMM & to optimize the tradeoff between their flexibility & the amount of training data to avoid the serious overfitting [24]. Accuracy of GMM -HMM system can further improved by discriminative training & augmenting the feature (e.g. MFCC) tandem or bottleneck features generated using MM [25, 26]. As an alternative of conventional HMM & GMM-HMM [23],SGMM & subspace GMM [27], DNN [28, 29, 30] convolutional MM [31, 32, 33, 30], RNN[34, 35] has been proposed and matter of deep investigation. To solve Hindi LVCSR Problem various DNM acoustic modeling should be tested and some novel acoustic modeling should be proposed.

Except acoustic modeling, language modeling has great impact on ASR performance. Language modeling is used to concise the multiple text hypothesis and determining the final output [36]. Deep learning & recurrent NN (RNN) have fueled language modeling research in part few years. Simpler models , such as N- grams , only use short history of previous words to predict the next word , they still a key component to high quality , low perplexity LM [37]. Most recent work on large scale LM has shown that RNNs are great in combination with NGrams , but worse when consider in isolation [38, 39, 40, 41, 42, 43, 44].

Table 1 describes the detailed significant work comparison in the field of Continuous Hindi Speech Recognition. We took only those works which were reported after the year 2010. Researchers do their

work either self-created dataset or Mandi dataset or TIFR, Mumbai dataset. Aggarwal & Dave reported 8.0% WER for 600 words vocabulary dataset in 2011. In the same series, various work was reported with different architectures. Recently, Kumar Aggarwal reports the 15.6% WER for TIFR, Mumbai dataset using TDNN based acoustic modeling. Still, there are huge possibilities to explore the DNN architectures for Hindi speech recognition.

3. Challenges and Issues in ASR

TVoice is the medium through which we pass our knowledge & opinion to others. Though there have been lots of developments made in the area of speech recognition, still some barriers are there to remove to notice present adoption in speech recognition. Seeing the progress, we can say that there is fast development in the field of speech recognition, but in general, it is still to be acquired in our day-to-day life. To comprehend the full capabilities of automatic speech recognition, there are hidden challenges that are yet to be discovered. Some features that affect the ASR are:-

- Speaking Style: ASR accuracy depends on like how is your speech tone, the rate at which you are speaking, the pitch of your voice, and phoneme production. Speech tone, if lowered, can't be listened to by ASR. Speech pitch creates difficulty in speech, and ASR finds it challenging to identify what word we are speaking.
- Speaker characteristics: Divergence in speech depends on the speaker's features like age, sex, and variation in speech caused by emotion, stress, and mental space.
- Environment: Environment in which ASR is set in, acts as one of the most effective factors for ASR. The environment includes background noise, room acoustics, and channel conditions.

Table 1. Previous work comparison on Hindi Speech Recognition

Author	Workon	Language	FeatureExtraction	Classification	LanguageModeling	Duration	Dataset	WER(%)
2011[45]	Continuous SpeechRecognition	Hindi	HetrogeneousRASTA-PLP+ MFCC+4GC	GMM	n-gram	600words ~1hour	self-created	8.0
2012[46]	NoisyContinuousSpeechRecognition	Hindi	RASTA+PLP+MFCC+LDA	GMM+Hybrid(Rover)	n-gram	~1hour	self-created	11.1
2013[47]	Continuous SpeechRecognition	Hindi+Marathi	MFCC	SubspaceGMM	n-gram	10hours	Mandi	24.0
2014[48]	Continuous SpeechRecognition	Hindi	WERBC	HMM	n-gram	2.5hours	TIFR,Mumbai	18.52
2015[49]	Continuous SpeechRecognition	Hindi	MFCC	HMM	n-gram	2.5hours	TIFR,Mumbai	12.6
2015[50]	Continuous SpeechRecognition	Hindi	LDA+MLLT+SAT	DNN-HMM	-	78hours	Mandi	18.0
2016[51]	Continuous SpeechRecognition	Hindi+Tamil+ Bangali+Assamies	MFCC	CNN	–	2-20hours	Mandi	14.51
2016[52]	NoisyContinuousSpeechRecognition	Hindi	Wavelet +Harmonic	HMM	–	2.5hours	TIFR,Mumbai	24.05
2017[53]	Continuous SpeechRecognition	Hindi+Tamil +Kanada	MFCC	DNN-HMM	–	10hours	Mandi	12.05
2018[54]	Continuous SpeechRecognition	Hindi	MFCC	GMM-HMM	RNNLM	2.5hours	TIFR,Mumbai	15.6
2019[55]	Continuous SpeechRecognition	Hindi	FBANK	CNN-BLSTM	n-gram	40hours	SpeechOcean	17.8
2019[56]	Continuous SpeechRecognition	Hindi	MFCC	TDNN	n-gram	2.5hours	TIFR,Mumbai	15.9

There is a large group of divergence in speech signals one should consider while creating an ASR. Every speaker has a unique vocal tone. Even when we try to speak a word twice, little difference in a voice still occurs. Considerate divergence is noticeable due to environmental and speaking conditions. Another big challenge for ASR is to tackle the mismatch issue where you speak does not match with word yielded by the system. Another problem in speech recognition seen during this development phase of ASR is controlling spectral and temporal mutability of speech. In earlier days, it was seen that

controlling the immeasurable mutability caused the failure in speech recognition. So it is very necessary that we should have a proper tool for handling the variability in automatic speech recognition systems.

4. Available Speech Corpus Details

The serious effort has been made by the Indian Language Technology Proliferation & Deployment Center for creating speech corpus for Indian languages. In this series, the Hindi speech dataset [57] was created and freely available via TDIL 1 for research purposes. The total duration of this dataset is around5.5 hours. This dataset is a speaker-independent dataset containing speech utterances of various speakers of different age groups.

The continuous Hindi speech dataset was developed by TIFR, Mumbai, in 1998 [58]. The total duration of this dataset is around 2.5 hours. This limited vocabulary Hindi speech dataset contains the speech utterances of 100 speakers. Each speaker utters ten sentences out of which two sentences were remains the same. In this way, 1000 sentences were recorded via male and female speakers of different age groups. The dataset was recorded in a quiet environment on 16 kHz sampling frequency.

Except for these, there are several Hindi speech datasets available but on paid bases. In this series, Linguistic Data Consortium (LDC)2 and Speechocean3 provide dataset for Large Vocabulary Continuous Speech Recognition (LVCSR) with a huge amount of subscription fee. SpeechOcean provides 308 hours of Hindi speech utterances. This dataset contains the speech utterances of 200 different speakers in which 108 male and 92 female speakers. These speakers belong to different age groups. The dataset was recorded in a quiet environment.

5. Acoustics Modeling Techniques

It is there depicts the connection between a linguistic unit and an audio signal. Acoustic modeling requires a group of sound recordings and their related transcription to understand the speech recognition system.

5.1. Statistical Method

These are the acoustic modeling methods, and these are one of the most used and accessible ways in the speech recognition system. Speech recognition system needs to transform speech signal into text which can be done statistically as following: - Suppose a group of acoustic observation

O = (O1,O2,O3,.....,On) and sequence of word W = (W1,W2,W3,...,Wn), so maximum probability

$$w = \text{argmax}P(W/O) = \text{argmax}(P(W) * P(O/W))/P(O) \quad (1)$$

This shows the most probable word sequence with the use of Bayes rule. Here P(O) can be neglected because it is free of sequence. So, W = argmax P(W) ∗ P(O/W)

5.1.1. Hidden Markov Model: It is a statistical method that can be used for ASR. It includes a group of states connected through transitions in this some states are hidden, and those states are not accessible by the observer, but speech vector sequence can be achieved using probability density function for each state. Hidden Markov model can be depicted as a hidden state function as in the figure 1. An HMM is defined by the following properties:-

A group of state S = (S1,S2,S3,.,Sn), so transition probability A = (a11,a12,.ann) where aij = transition probability from state I to j. Initial state distribution Pi = P[S(O) = Si],i = 1,2,3,.,N, Hidden Markov Model is called to be complete if A,B,Pi are given, and this model can be denoted by Y= (A,B,Pi). HMM is based on the assumption that voice signals consist of stationary little time segments. Since the Hidden Markov Model is good at picking up the changeability in speech, that's why they are commonly used in ASR.

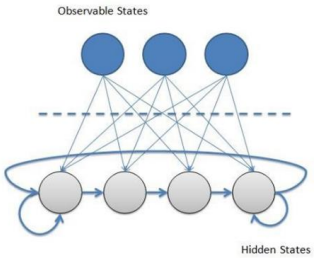

Fig 1. Basic Block Diagram of Hidden Markov Model

We know that HMM permits transitions from one state to another, but in SRS transitions are not permitted to starting states. This type of HMM is known as the left-right network. This HMM has the ability to model speech signals that alter their nature in time. An HMM is considered to be a phoneme for an ASR.

5.2. Deep Neural Network (DNN)

It is a type of artificial neural that which consists of many layers, including the input and layers layer. A deep neural network uses mathematical formula convert input into the output of linear or non-linear connection. DNN uses the probability of output to go through different layers. Experiments show that the modeling potential of deep neural network increases while including hidden layers, which consists of large numbers of neurons. It has the ability to manage the complexity of layers. In DNN, nodes of different layers learn from a variety of qualities which is created from the output of different layers. As you go deep in the DNN, the next layer becomes more and more complex. While moving through the layer, DNN adapts to identify the relationship between characteristics and output of a certain level.

5.2.1. Convolution Neural Network (CNN): CNN is a type of neural network in this model consists of a convolution and pooling layer, and they are put on each other. In the convolution layer, weight is given while the pooling layer handles the output of the convolution layer. The pooling layer minimizes the data rate of the convolution layer. Pooling schemes are used to share weight together, which causes the invariance of CNN. Some people say invariance in CNN is not adequate for complex pattern recognition, but it still shows usefulness when it comes to image recognition and computer vision. The properties of CNN show that it can also be used in speech recognition.

5.2.2. Recurrent Neural Network (RNN): It is a type of neural network that can be used in unsupervised learning. RNN can be used where the depth of the data sequence is equal to length. RNN is build using the same group of weight in pyramid-structure using recursion. It is useful to upcoming data order using earlier data. It is very convincing to use to model data order of text or speech. Since RNN is difficult to train so they are not used at large scale in past. Using

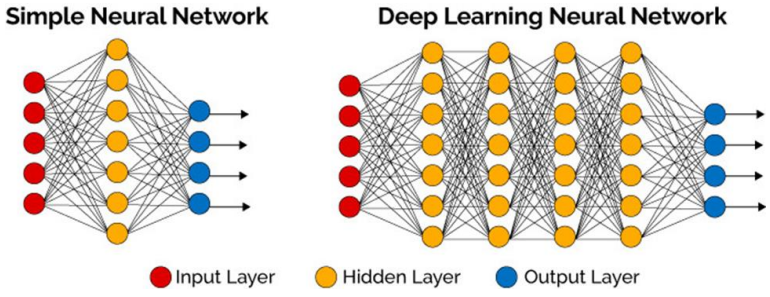

Fig 2. DNN Architecture.

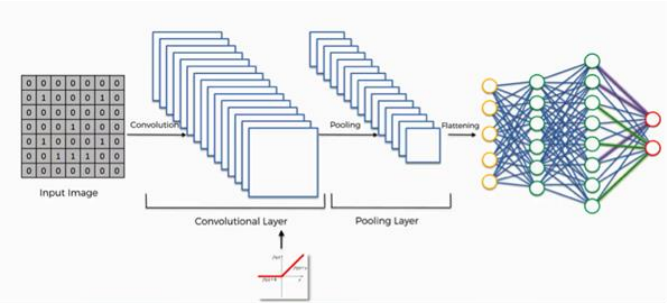

Fig 3. Architecture of CNN

Hessian free Optimization removed this difficulty. This optimization technique helped RNN to be able to generate sequential text characters.

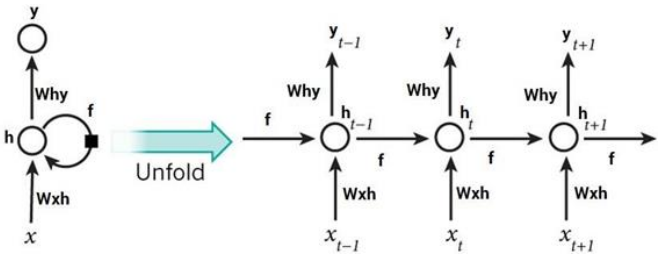

Fig 4. Architecture of Recurrent Neural Network

6. SAPI Methodology

Speech recognition technology is still a relatively new technology, so not many developers are familiar with it. While understanding the fundamental functions of speech synthesis and speech recognition takes only a few minutes (after all, most people learn to speak and listen by the age of two), there are subtle and powerful capabilities provided by computerized speech that developers will want to understand and use. Above all, voice technology does not always live up to the high expectations of users who are accustomed to authentic human-to-human conversation. For successful usage of voice input and output in a user interface, as well as comprehending some of the sophisticated features of the Speech API, it's vital to grasp the limitations as well as the benefits. It's also important to know what speech technology can do and what it can't do.

Understanding the capabilities and limitations of speech technology is particularly crucial for developers when deciding whether to employ speech input and output in a given application. Speech API, frequently abbreviated as SAPI, is a Microsoft open-source speech development kit that we are employing in our project.

The Speech Application Programming Interface, or SAPI, is a Microsoft API that enables the use of speech detection and synthesis within applications. Applications that run on Windows. The Speech API can be thought of as a kind of interface or middleware that sits between programmes and speech engines (recognition and synthesis). We're utilising SAPI 5.1, which is part of Microsoft's Speech Development Kit version 5.0. SAPI 5.1 is the latest version of SAPI.

This version was released as part of the Speech SDK version 5.1 in late 2001. The API now has automation-compliant interfaces that may be used with Visual Basic, scripting languages like JScript,

and managed code [10]. Windows XP included with this version of the API and TTS engines. Office 2003 and Windows XP Tablet PC Edition.

7. Future Scope

7.1. All speech processing applications, such as ASR and TTS, require the handling of real (unrestricted) text, either directly or indirectly. To handle the massive amounts of speech and text (transcript) data, a web application was created utilising JavaScript and Node.js that efficiently processes vast amounts of audio recordings and their transcriptions, as well as doing collaborative human correction of the ASR transcripts. The web app was built with a client-server architecture and a MongoDB database to function in real-time. The web software used a REST API to handle simultaneous logins from several users, each with their own user ID, password, and language ID. All wav audio files from the local machine must be loaded into the web app by the signed in user . The transcripts of the related audio files would then be shown to the user. Any changes to the transcripts were saved to the server as txt files. Users may also use the software to convert numbers to words and expand abbreviations by just picking the text and requesting the conversion from the server. The server was updated with information on each manual edit, and a change record was kept, which could be used to provide incentives to volunteers depending on the editing pipeline updates.

7.2. Speech recognition was made possible with the creation of a web interface that allowed users to record their voice and have it recognised and transcribed into text by the server. This necessitated the capture of 16 kHz mono channel audio from each device that hosted the web app. This was accomplished by removing extraneous data points from the recorded voice vector and down sampling the data. There are pauses or silent intervals in speech files. Because these intervals do not carry any speaker information, the inclusion of non-speech periods in speech files needs voice activity recognition. The web interface was created to perform voice activity detection, also known as speech activity detection, to detect the presence or absence of human speech, and to facilitate efficient speech processing by deactivating some processes during non-speech intervals of an audio session to avoid unnecessary coding and transmission of silence packets in VoIP applications, saving on computation and network bandwidth.

This was accomplished by examining the greatest amplitude reached inside a given FFT window using the time-domain data from the voice vector. The vector contained speech if it above an empirically defined threshold; else, it was silence that could be ignored.

8. Conclusion

This paper is related to Automatic Speech Recognition (ASR) field, the witness of remarkable improvement over the last few years. Deep learning architectures are the main reason for ASR evaluation. Deep learning techniques tremendously improve ASR performance, but these architectures are data-hungry. There are more than 5000 languages in India that do not have such a huge amount of resources. Hindi is one of them. The lack of freely available Large Vocabulary Hindi speech datasets is the major hurdle for developing Hindi ASR. In this paper, we investigate the various work available for Hindi ASR. This article provides a detailed analysis of state-of-the-art Hindi ASR systems. This will helps the new researchers by understanding the existing work and their limitations. As discussed in previous sections, DNN techniques are not properly investigated in Hindi ASR filed. Except for acoustic modeling, language models are also not properly examined, which can be examined in the near future. This article attempts to give a detailed review of Hindi ASR and find the existing gaps for future enhancements.

References

[1] Li J, Deng L, Gong Y and Haeb-Umbach R 2014 IEEE/ACM Transactions on Audio, Speech, and Language Processing22 745–777
[2] Deng L, Wang K, Acero A, Hon H W, Droppo J, Boulis C, Wang Y Y, Jacoby D, Mahajan M, Chelba C et al. 2002 IEEE Transactions on Speech and Audio Processing 10 605–619
[3] Erzin E, Cetin A E and Yardimci Y 1995 1995 International Conference on Acoustics, Speech, and Signal Processing vol 1 (IEEE) pp 417–420
[4] Irino T and Patterson R D 1997 The Journal of the Acoustical Society of America 101 412–419
[5] Kleinschmidt M and Gelbart D 2002 Seventh international conference on spoken language processing
[6] Bourlard H A and Morgan N 2012 Connectionist speech recognition: a hybrid approach vol 247 (Springer Science & Business Media)
[7] Hermansky H, Ellis D P and Sharma S 2000 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100) vol 3 (IEEE) pp 1635–1638
[8] Gr´ezl F, Karafia´t M, Konta´r S and Cernocky J 2007 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07 vol 4 (IEEE) pp IV–757
[9] Yu D, Deng L and Dahl G 2010 Proc. NIPS Workshop on Deep Learning and Unsupervised Feature Learning
[10] Stern R M, Acero A, Liu F H and Ohshima Y 1996 Automatic Speech and Speaker Recognition (Springer) pp 357–384
[11] Deng L, Acero A, Plumpe M and Huang X 2000 Sixth International Conference on Spoken Language Processing
[12] Xiao X, Li J, Chng E S and Li H 2012 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (IEEE) pp 4305–4308
[13] Cui X, Xue J and Zhou B 2009 2009 IEEE Workshop on Automatic Speech Recognition & Understanding (IEEE) pp 136–140
[14] Gemmeke J F and Virtanen T 2010 2010 IEEE International Conference on Acoustics, Speech and Signal Processing (IEEE) pp 4546–4549
[15] Raj B, Virtanen T, Chaudhuri S and Singh R 2010 Eleventh Annual Conference of the International Speech Communication Association
[16] Gemmeke J F, Virtanen T and Hurmalainen A 2011 IEEE Transactions on Audio, Speech, and Language Processing 19 2067–2080
[17] Maas A, Le Q V, O'neil T M, Vinyals O, Nguyen P and Ng A Y 2012
[18] W¨ollmer M, Zhang Z, Weninger F, Schuller B and Rigoll G 2013 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (IEEE) pp 6822–6826
[19] Virtanen T 2007 IEEE transactions on audio, speech, and language processing 15 1066–1074
[20] Gemmeke J et al. 2012 Proceedings Interspeech 2012 1–4
[21] Weninger F, W¨ollmer M, Geiger J, Schuller B, Gemmeke J F, Hurmalainen A, Virtanen T and Rigoll G 2012 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (IEEE) pp 4681–4684
[22] Grais E M and Erdogan H 2013 Interspeech pp 808–812
[23] Vegesna V V R, Gurugubelli K, Vydana H K, Pulugandla B, Shrivastava M and Vuppala A K 2017 International Conference on Mining Intelligence and Knowledge Exploration (Springer) pp 189–197
[24] Saon G and Chien J T 2011 IEEE Transactions on Audio, Speech, and Language Processing 20 43–54
[25] Malioutov D M, Sanghavi S R and Willsky A S 2010 IEEE Journal of Selected Topics in Signal Processing 4 435–444
[26] Ji S, Xue Y and Carin L 2008 IEEE Transactions on signal processing 56 2346–2356
[27] Mohan A, Rose R, Ghalehjegh S H and Umesh S 2014 Speech Communication 56 167–180
[28] Povey D, Peddinti V, Galvez D, Ghahremani P, Manohar V, Na X, Wang Y and Khudanpur S 2016 Interspeech pp 2751– 2755
[29] Yoshioka T and Gales M J 2015 Computer Speech & Language 31 65–86
[30] Abraham B, Umesh S and Joy N M 2016 INTERSPEECH pp 3037–3041
[31] Xiong W, Wu L, Alleva F, Droppo J, Huang X and Stolcke A 2018 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP) (IEEE) pp 5934–5938
[32] Abdel-Hamid O, Mohamed A r, Jiang H, Deng L, Penn G and Yu D 2014 IEEE/ACM Transactions on audio, speech, and language processing 22 1533–1545
[33] Saon G, Kuo H K J, Rennie S and Picheny M 2015 arXiv preprint arXiv:1505.05899
[34] Graves A and Jaitly N 2014 International conference on machine learning pp 1764–1772
[35] Mohamed A r, Seide F, Yu D, Droppo J, Stoicke A, Zweig G and Penn G 2015 2015 IEEE Workshop on 12 Automatic Speech Recognition and Understanding (ASRU) (IEEE) pp 78–83
[36] Bellegarda J R 2004 Speech communication 42 93–108
[37] Jozefowicz R, Vinyals O, Schuster M, Shazeer N and Wu Y 2016 arXiv preprint arXiv:1602.02410
[38] Saon G and Chien J T 2011 IEEE Transactions on Audio, Speech, and Language Processing 20 43–54
[39] Mikolov T, Karafia´t M and Burget L 2010 Eleventh annual conference of the international speech communication association pp 1045–1048

[40] Mikolov T and Zweig G 2012 2012 IEEE Spoken Language Technology Workshop (SLT) (IEEE) pp 234–239
[41] Chelba C, Mikolov T, Schuster M, Ge Q, Brants T, Koehn P and Robinson T 2013 arXiv preprint arXiv:1312.3005
[42] Williams W, Prasad N, Mrva D, Ash T and Robinson T 2015 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (IEEE) pp 5391–5395
[43] Shazeer N, Pelemans J and Chelba C 2015 Proceedings Interspeech 2015 2015 1428–1432
[44] Chen X, Liu X, Gales M J and Woodland P C 2015 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (IEEE) pp 5411–5415
[45] Aggarwal R K and Dave M 2013 Telecommunication Systems 52 1457–1466
[46] Aggarwal R K and Dave M 2012 International Journal of Speech Technology 15 165–180
[47] Mohan A, Rose R, Ghalehjegh S H and Umesh S 2014 Speech Communication 56 167–180
[48] Biswas A, Sahu P K and Chandra M 2014 Computers & Electrical Engineering 40 1111–1122
[49] Sharan S, Bansal S and Agrawal S 2018 Speech and Language Processing for Human-Machine Communications (Springer) pp 91–97
[50] Biswas A, Sahu P, Bhowmick A and Chandra M 2016 IETE Journal of Research 62 129–139
[51] Mandal P, Jain S, Ojha G and Shukla A 2015 Sixteenth Annual Conference of the International Speech Communication Association
[52] Abraham B, Umesh S and Joy N M 2016 INTERSPEECH pp 3037–3041
[53] Biswas A, Sahu P K and Chandra M 2016 IET Signal Processing 10 902–911
[54] Abraham B, Seeram T and Umesh S 2017 INTERSPEECH pp 2158–2162
[55] Passricha V and Aggarwal R K 2019 Journal of Intelligent Systems 1
[56] Kumar A and Aggarwal R 2020 Advances in Data and Information Sciences (Springer) pp 425–432
[57] Jha G N 2012 Language Resources and Evaluation Conference
[58] Samudravijaya K, Rao P and Agrawal S 2000 Sixth International Conference on Spoken Language Processing