Kannada Speech to Text Conversion Using CMU Sphinx

Shivakumar K.M, Aravind K.G, Anoop T.V
Dept. of Computer Science,
Amrita Vishwa Vidyapeetham University, Mysore campus,
Amrita University, Mysuru, India

shivadvg19@yahoo.com

DeepaGupta
Dept. of Mathematics,
Amrita School of Engineering,Bengaluru campus
Amrita Vishwavidyapeetham University,
Amrita University, Bengaluru, India
g_deepa@blr.amrita.edu

***ABSTRACT*:This paper investigates the complex problem of speech to text conversion of Kannada Language. We propose a novel Kannada Automated Speech to Text conversion System (ASTC). We train and test the Speech Processing System using CMUSphinx framework. CMU Sphinx is dynamic in nature with support for other languages along with English. We train the Acoustic model for Kannada speech with 1000 general spoken sentences and tested 150 sentences. We build our system utilizing features available in CMU Sphinx, thus showcasing the conceivable flexibility of this framework for Kannada voice to text conversion. In this paper, Kannada sentences with four to ten word length is researched. The speech conversion system permits ordinary people to speak to the computer in order to retrieve information in textual form. The number of alphabets in Kannada are 52. The system investigates extensibility of recognizing all letters and morphological variants of spoken Kannada words.**

Keywords: ***Speech to text conversion Speech Recognition CMUSphinx-4 Acoustic modeling Phoneme Language model***

I. INTRODUCTION

Natural Language Processing is an area under artificial Intelligence intended to accept process and manipulate the human language. It helps to model cognitive behavior of human mind to process natural language. The NLP is used to identify the natural language text and speech. In this process the mathematical, computational and linguistic knowledge has to be integrated to develop an application which serves as an aid to human beings in improving their task of perceiving their day to day information. NLP is used in Speech, Text, Sentiment Analysis and other applications. Application of Machine Learning Algorithm on archived natural language data such as mono-lingual or bi-lingual text or speech corpus acquires the techniques to process the natural language. Processing a natural language involves identifying a given language font, word, phrase or a sentence. This task opens a wide avenue of research to carryout in developing language processing tools such as dictionaries, wordnet, parts of speech taggers, morphological analyzers, machine translation systems and automatic speech recognizers. Speech processing is one of the application area of NLP which involves encoding and decoding the audio signals uttered by human being to produce a letter, word or sentence. Speech Processing uses Hidden Markov Models to extract features and other statistical models. These models are used to encode speech to its textual form and used in decoding to represent in textual form. Speech Recognition Systems which now rely on cache language models. These models extract the language pattern and the order of arrival of word or speech sequences and classify them based on their probability of occurrence in real time. These probabilistic estimations produce more accurate results when we are testing with unknown data or speech sequence. Speech to Text Conversion had a wide range of application in human language processing. It is useful for business people to business meeting notes. Speaker identification and for deaf people is helpful to know the information.

In this paper we present our findings in processing an Indian language Kannada being spoken in the state Karnataka, India. It is an effort

to have a human computer dialogue system in any local language. We describe our experience to create and develop Kannada Language acoustic model using CMU Sphinx-4 speech recognition system.

II. Related Works

Yang Liu et.al (2003) proposed a method to find disfluencies in a speech, they used acoustic-prosodic features, language model (LM), part-of-speech (POS) based LM, and rule-based knowledge. They trained the system with switchboard-I corpus with speech conversations and labeled the disfluencies. They obtained the accuracy in identifying disfluencies with precision rate of 68% and recall rate of 46.41%. Matthias Honal et.al (2003) proposed a method for finding disfluencies in a speech. Statistical machine translation approach was used in this process; the noisy channel adds noise to its input sentence for disfluencies and creates the sentence in the source language as output. They experimented with English and Chinese language. For English finding disfluency rate was 77.2% recall and 90.2% precision. They used Mandarian Chinese data.

Neema Mishra et.al (2013) combined methods like feature extraction, acoustic modelling and language modelling in Sphinx methodology for identifying disfluencies in Hindi language. The system was able to recognize the Hindi audio successfully. Satori et.al (2007) proposed a method to identify disfluency in Arabic speech. They used Sphinx methodology having acoustic model, Language model and Dictionary. The result obtained was 80% accuracy.Dana Dann´ells [5] proposed a work based on combinations of lexical, semantic, syntactic and acoustic Information for measuring disfluency rates and the data sample. Based on empirical evidence and theoretical findings, computing linguistic and non-linguistic features is a powerful approach in association with other attributes to Determine whether an utterance is disfluent and its signals are cognitive load.

Yulia Tsvetkov et.al (2013) proposed a method to handle disfluencies, word phrases and self-interruption points in Cantonese conversational speech data. The methods used are Word Fragments Identification and Word Fragments Modelling along with Annotation of word fragments. They achieved an accuracy of 88% for automatic detection of disfluencies. Jordi Adell et.al (2006) proposed a preliminary analysis of pitch and segmental duration in repetitions and filled pauses. Experimental results shows that the hypothesis of pitch for filled pauses were generally lower than their contexts.

Matthew Lease et.al (2006) proposed a system which is able to predict Interruption Points by simply taking the union of Edit Word Detection and Filler Word Detection predictions. The methods used were Simple MDE annotations. On both types of input, the system was the top performer in the evaluation. The system uses the approach of automatically detecting sentence boundaries to improve its accuracy. Prashanth Kannadaguli et.al (2015) proposed an automatic phoneme recognition system based on Bayesian Multivariate Modeling. Phoneme models were built by using stochastic pattern recognition and acoustic phonetic schemes to recognize phonemes. They have used 15 Kannada phonemes to train and test these models. The performance analysis of these models in terms of Phoneme Error Rate (PER) for Likelihood and Posterior were obtained for 15 phonemes.

Kallirroi Georgila et.al (2010) proposed a model for speech disfluency detection based on conditional random fields (CRFs) using the Switchboard corpus. It show that a technique for detecting speech disfluencies based on ILP significantly outperforms CRFs. In terms of F-score and NIST Error absolute improvement of Integer Linear Programming (ILP) over CRFs exceeds 20% and 25% respectively. Hemakumar G et.al (2011) proposed a system for creating Acoustic Phonetic Characteristics of Kannada Language, 445 simple and most frequently used Kannada words were selected and used to measure the features of phonemes in continuous speech. They modeled the comparative analysis of short and long vowels durations in speech.

III. Methodology of Proposed Work

The applications to encode and decode the speech data for English are supported by huge database of language models. Availability of this database enhanced the performance of speech recognition accuracy for English. This is not the same for other Asian or Indian Languages. To generate an Acoustic model for Indian languages is a challenging task. Choosing a tool and configuring the system to work for Indian language requires a series of configuration steps with open source tools such as SIP and CMU Sphinx. Little work has been done to work with these tools for recognizing Indian Language speech compared to English language.

3.1 *An ASR System like Sphinx-4 uses three types of language-dependent models:*

Acoustic model for Kannada represents statistical range of possible audio representations for Kannada Language phonemes. *A pronunciation dictionary* specifying how each word is pronounced in terms of the phonemes in the acoustic model.

A language model (LM) models pattern probability of occurrence of words. This is normally customized for domain specific application. Every word in the language model must be in the pronunciation dictionary.In our Acoustic model training we created above models according to CMU Sphinx framework specifications for new languages.

3.2 Data Preparation

We used publicly available speech corpora for Kannada created by IIIT Hyderabad. The corpus contains 1000 sentences related to general information about Karnataka. The corpus files were of single speaker. We created speech sample of the same sentences and converted into two user speech data. The corpus comprise 5000 words in which 2112 are unique words. These words are span through the usage of 36 phonemes. The recorded samples were rechecked to ensure the utterances were recorded efficiently.

	No. of Files	No. of Sentences	No. of Words
Train	1000	1000	6694
Test	151	151	1036

Table1: Speech Corpus Statistics

3.3 Language model

The aim of the language model is to produce accurate value of probability of a word. A language model represents the structural constraints available in the language to generate the probabilities. In language model, two different words have similar sounding phone. The LM also specifies what are the valid words in the language and their arrival sequence in the speech data. The generated model for Kannada plays a vital role in generating context-dependent and context-independent acoustic model generation for Kannada speech sample.

3.4 Phoneme set:

Kannada spoken language uses 52 alphabets. These alphabets are transformed into 36 phonemes which are written using English alphabets according to Sphinx-4 specifications to represent phonemes for new language. Based on these phonemes Kannada word dictionary is prepared.

3.5 Dictionary Preparation:

Dictionary should comprise the words which are constituents of available phoneme set. Every word should be a subset of available phoneme group of the word. For each token T should be a subset group of basic phonemes. We are using 36 phonemes for Kannada Speech Processing System.

$$T \subseteq S \{p_1, p_2 \ldots . p_n\}$$

Where S is a subset of phonemes of a given language. T is any word or token of pronunciation dictionary. We created 2112 Kannada tokens in our dictionary.

3.6 Feature Extraction

The main features of our system are frequency of the waveform of the speech data. The system found the upper frequency as 6800 and lower frequency as 130 for the Kannada speech. It uses Nfilt parameter value of 52 and discrete cosine transformation (DCT) to extract transformations. The speech sample features also considers the age of the speaker as a parameter. Speaker-dependent information exists both in the spectral envelope and in the supra-segmental features of speech. This individual information can be further classified into temporal and dynamic features. Speaker identification/verification methods can be divided into text-dependent and text-independent methods. We are using text dependent feature extraction.

3.7 Acoustic model:

Acoustic model creation for the given speech data in Sphinx framework is done by sphinx train module. It uses built in HMM and Viterbi algorithms and modules to train the system by itself. The output of this phase is written in configuration file. This configuration file describes the components and their respective values to be used by the speech processing system. The parameters written in configuration file are used by the system to run the decoder which generates the acoustic model for a given language.

3.8 Acoustic-Phonetic features of Kannada

The acoustic-phonetic of Kannada differs from the European languages. The Kannada alphabet consists of 14 vowels, 25 stop consonants and 9 non-stop consonants. The stop consonants are ordered in a systematic manner in most of the Indian language and this order may suggest ideas for developing a recognition system. The pronunciation of Kannada alphabets and words are to be modeled with the help of English letter based phoneme set. Since all letters in Kannada are

transcribed into their respective phonemes using English letters and special characters. Here we described characteristic of Kannada Vowel and Consonants. In Table 2. It has two sections: First section consists of vowels and second section consists of consonants.

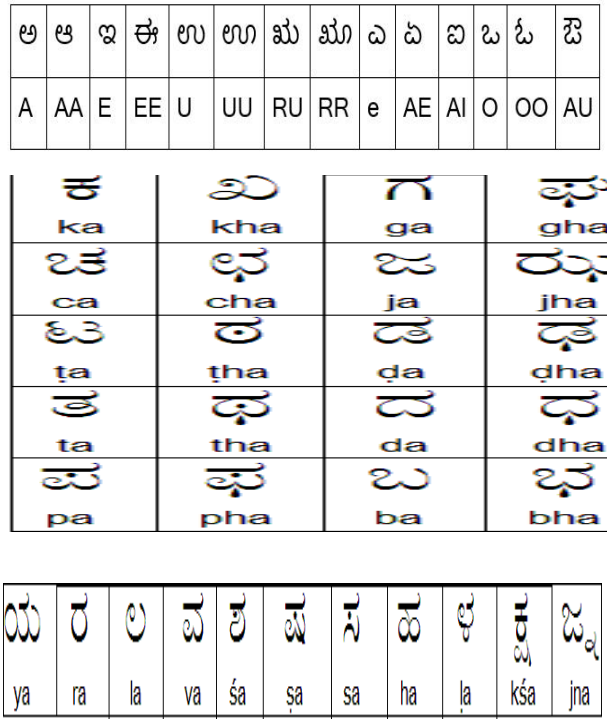

Table 2: Kannada Letters

Table 2 suggest that we can model Kannada basic fonts and words with available phone set for English which is the only supporting version by sphinx.

3.9 Sphinx Train

Sphinx-4 is used to convert speech recordings into Text. It also helps to identify the speakers, adapt models, and align existing transcription to audio. It is a speech recognition library in Java. *Sphinx Train* creates an acoustic model which is manipulated by Sphinx-4. This acoustic model is important for any language for speech recognition. Once data is trained, the database will be created. The next step is to create speech recognition files by running the sphinx train.

IV RESULTS:

4.1 Feature Extraction Parameters

The objective of feature extraction is to encode each and every utterance and group them based on their characteristic features. To classify a speech signal into its respective group we take certain set of features and these play a major role in grouping the acoustics to their respective classes.

Features	Values
Lowerf	130
Upperf	6800
Nfilt	52
Transform	Dct
Lifter	22
Feat	1s_c_d_dd
Agc	None
Cmn	Current
Varnorm	No

Table3: Feature extraction details

4.2 Experiments with Different Number of Trainings

We started our training and testing with 150(training)/15(testing), 300/30,500/50,700/70 and 1000/150 sentences audio files. The experiment shows increase in accuracy level as the number of training files increased. Initially Thousand sentences of Kannada language are recorded and trained with different. Testing of randomly chosen one fifty sentences are made and the results are given in figure 1 and figure 2.

4.2.1Context Dependent Model:

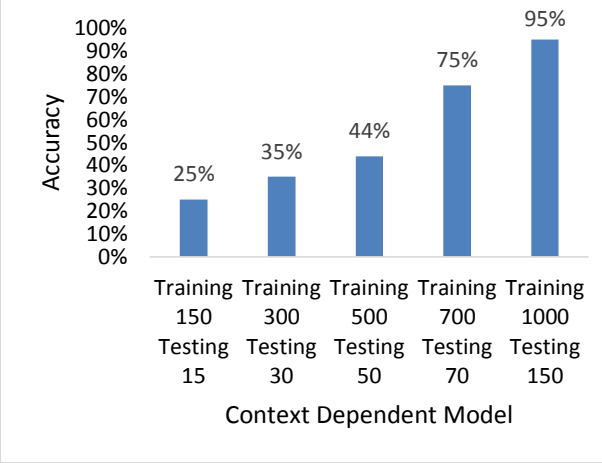

Figure1: Context Dependent Model

Context Dependent Model is used to train and test the single user speech. The above diagram shows the increase in accuracy level with increase in training set.

4.2.2Context Independent Model:

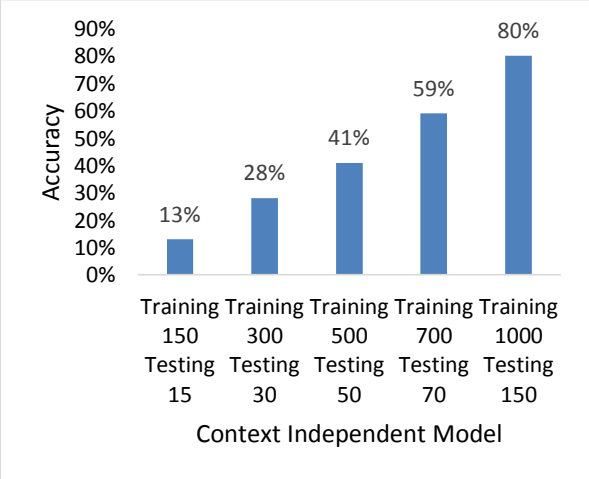

Figure2: Context Independent Model

Context Independent Model uses multiuser speech files and requires more training data compared to context dependent model. The overall accuracy level in context independent model based recognition is less compared to context dependent model based recognition and text conversion. The Figure 3 and Figure 4 shows the experiment results.

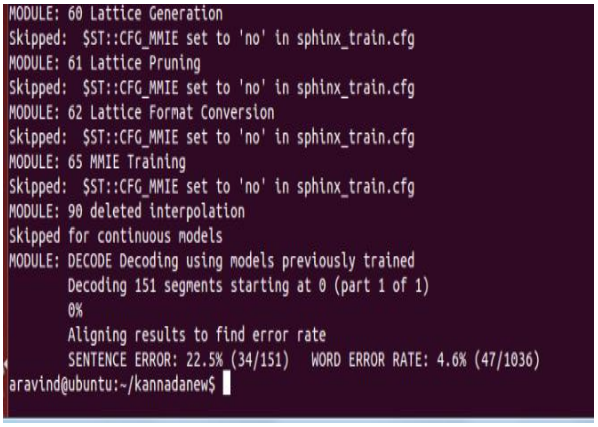

Figure 3: Accuracy of the Context Dependent Model

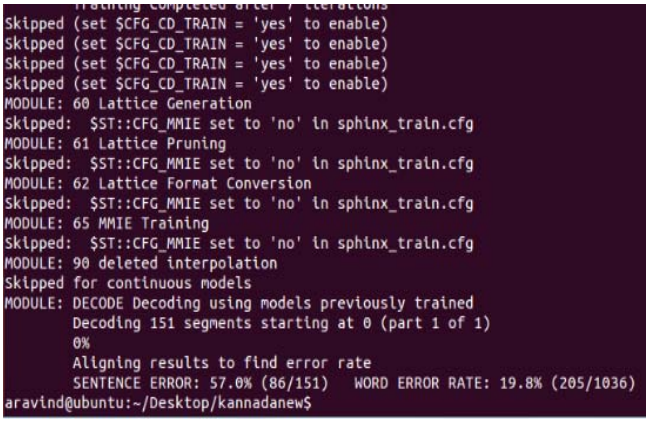

Figure 4: Accuracy of the Context Independent Model

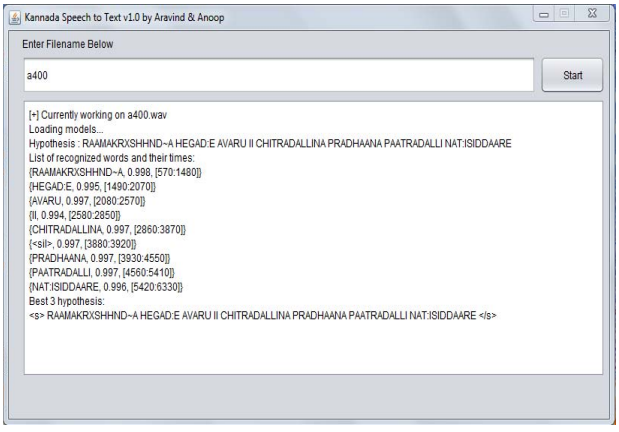

Figure 5: Sample Output

The figure5 is a sample output of speech to text conversion of a given file.

IV. CONCLUSION

We experimented speech to text conversion system for Kannada Language with

limited speech corpus of thousand sentences. These sentences are used by the system to generate acoustic model. The decoder generates the sentence and word error rate for the given speech sample. This helps to measure the accuracy of speech to text conversion system. We are planning to create domain specific speech corpus and implement it on Sphinx framework to work on speaker identification and Live speech Recognition.

REFERENCES

[1] Liu, Y., Shriberg, E., & Stolcke, A. (2003, September). Automatic disfluency identification in conversational speech using multiple knowledge sources. In *INTERSPEECH*.

[2] Honal, M., & Schultz, T. (2003, April). Correction of disfluencies in spontaneous speech using a noisy-channel approach. In *INTERSPEECH*.

[3] Mishra, N., Shrawankar, U., & Thakare, V. M. (2013). An Overview of Hindi Speech Recognition. *arXiv preprint arXiv:1305.2847*.

[4] Satori, H., Harti, M., & Chenfour, N. (2007). Introduction to Arabic speech recognition using CMUSphinx system. *arXiv preprint arXiv:0704.2083*.

[5] Dana Dann´ells, "Disfluency detection in a dialogue system".

[6] Tsvetkov, Y., Sheikh, Z., & Metze, F. (2013, May). Identification and modeling of word fragments in spontaneous speech. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*(pp. 7624-7628). IEEE. .

[7] JAdell, J., Bonafonte, A., Escudero, D., & Informatics, D. (2006, May). Disfluent speech analysis and synthesis: a preliminary approach. In *in Proc. of 3th International Conference on Speech Prosody*.

[8] Lease, M., Johnson, M., & Charniak, E. (2006). Recognizing disfluencies in conversational speech. *Audio, Speech, and Language Processing, IEEE Transactions on*, *14*(5), 1566-1573.

[9] Kannadaguli, P., & Thalengala, A. (2015, February). Phoneme modeling for speech recognition in Kannada using Hidden Markov Model. In *Signal Processing, Informatics, Communication and Energy Systems (SPICES), 2015 IEEE International Conference on* (pp. 1-5). IEEE.

[10] Georgila, K., Wang, N., & Gratch, J. (2010, September). Cross-domain speech disfluency detection. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue* (pp. 237-240). Association for Computational Linguistics.

[11] Hemakumar, G. (2011). Acoustic Phonetic Characteristics of Kannada Language. International Journal of Computer Science Issues, 8(2).

[12] Huang, X., Alleva, F., Hon, H. W., Hwang, M. Y., Lee, K. F., & Rosenfeld, R. (1993). The SPHINX-II speech recognition system: an overview. *Computer Speech & Language*, *7*(2), 137-148.