**NLP Research: Bangla Speech and Phonetics using English Detection Models and Dictionaries**

* **Phoneme**: English or Latin word representations of atomic syllables and pronunciations, capitalization denotes hard sounds. IPA is an somewhat internationally accepted format for phoneme notation. English and Bangla have different sets of IPA
* **Hidden Markov Model**: HMMs are statistical models that can be used to describe the evolution of observable events that depend on unobservable internal factors. They provide an effective framework for modelling time-varying spectral vector sequences, and are great for LVCSR (Large Vocabulary Continuous Speech Recognition).
* **Mel Frequency Cepstral Coefficent**: MFCCs are audio signal components for identifying linguistic content based on utterance motions and calculated by the Inverse Fourier Transform of the Log (IFTL) of the estimated signal spectrum.
* **Soundex**: A phonetics distance algorithm that determines a four character hash group for syllables that are similar by sound and not spelling and sorts them by an efficient hash index.
* **PocketSphinx**: Speech recognition toolkit from Carnegie Mellon University that is Python wrapped and is mobile deployable. Has word dictionaries and phonetic models for various languages.
* **Kaldi**: Open-source speech data manipulation toolkit. Primarily built for C++ and also wrapped for Python. Low-level CMVN (Cepstral Mean Variance Normals) functions use BLAS or CUDA for efficiency.
* The phonemes detected using pocketsphinx via the ‘allphone-ci’ config are detected context independently.
* Convert latin phonemes or bangla unicode characters using python by character matching and phoneme mapping. Refer to the CMUBET table and the 'contrastive english bangla phonetic’ paper.
* Use G2P to build a dictionary ‘g2p-sec2sec --interactive’
* Run ‘sphinx\_lm\_convert -i model.lm -o model.lm.bin’ to generate the CMU binary language model.

1. Collect word list
2. Download espeak
3. Use espeak to create dictionary with embedded letter to sound rules
4. Convert dictionary in CMUSphinx-readable format.
5. Download text corpus
6. Download cmuclmtk
7. Follow doc cmuclmtk/doc/toolkit\_documentation\_no\_tables.html to create the language model.
8. Learn about building language model from inflectional languages.
9. Create/download morphological parser
10. Segment text on morphemes
11. Build morphological language model.

**CMUBET IPA Example Translation**

------ --- ------- -----------

AA ɑ odd AA D

AE æ at AE T

AH ʌ hut HH AH T

AO ɔ ought AO T

AW ɑʊ cow K AW

AY ɑɪ hide HH AY D

B b be B IY

CH ʧ cheese CH IY Z

D d dee D IY

DH ð thee DH IY

EH ɛ Ed EH D

ER ɜɹ hurt HH ER T

EY eɪ ate EY T

F f fee F IY

G ɡ green G R IY N

HH h he HH IY

IH i it IH T

IY ɪː eat IY T

JH ʤ gee JH IY

K k key K IY

L l lee L IY

M m me M IY

N n knee N IY

NG ŋ ping P IH NG

OW oʊ oat OW T

OY ɔɪ toy T OY

P p pee P IY

R ɹ read R IY D

S s sea S IY

SH ʃ she SH IY

SIL . silence \*wordend [ SIL ] wordstart\*

T t tea T IY

TH θ theta TH EY T AH

UH ʊ hood HH UH D

UW u two T UW

V v vee V IY

W w we W IY

Y j yield Y IY L D

Z z zee Z IY

ZH ʒ seizure S IY ZH ER

This program opens the audio device and waits for speech. When it detects an utterance,

it performs speech recognition on it.

-adchdr

Size of audio file header in bytes (headers are ignored)

-adcin Input is raw audio data

-agc Automatic gain control for c0 ('max', 'emax', 'noise', or 'none')

-agcthresh

Initial threshold for automatic gain control

-allphone

phoneme decoding with phonetic lm

-allphone\_ci

Perform phoneme decoding with phonetic lm and context-independent units only

-alpha Preemphasis parameter

-argfile

file giving extra arguments.

-ascale

Inverse of acoustic model scale for confidence score calculation

-aw Inverse weight applied to acoustic scores.

-backtrace

Print results and backtraces to log file.

-beam Beam width applied to every frame in Viterbi search (smaller values mean wider

beam)

-bestpath

Run bestpath (Dijkstra) search over word lattice (3rd pass)

-bestpathlw

Language model probability weight for bestpath search

-build\_outdirs

Create missing subdirectories in output directory

-cepdir

files directory (prefixed to filespecs in control file)

-cepext

Input files extension (suffixed to filespecs in control file)

-ceplen

Number of components in the input feature vector

-cmn Cepstral mean normalization scheme ('current', 'prior', or 'none')

-cmninit

Initial values (comma-separated) for cepstral mean when 'prior' is used

-compallsen

Compute all senone scores in every frame (can be faster when there are many

senones)

-ctl file listing utterances to be processed

-ctlcount

No. of utterances to be processed (after skipping -ctloffset entries)

-ctlincr

Do every Nth line in the control file

-ctloffset

No. of utterances at the beginning of -ctl file to be skipped

-ctm output in CTM file format (may require post-sorting)

-debug level for debugging messages

-dict pronunciation dictionary (lexicon) input file

-dictcase

Dictionary is case sensitive (NOTE: case insensitivity applies to ASCII characters

only)

-dither

Add 1/2-bit noise

-doublebw

Use double bandwidth filters (same center freq)

-ds Frame GMM computation downsampling ratio

-fdict word pronunciation dictionary input file

-feat Feature stream type, depends on the acoustic model

-featparams containing feature extraction parameters.

-fillprob

Filler word transition probability

-frate Frame rate

-fsg format finite state grammar file

-fsgctl

file listing FSG file to use for each utterance

-fsgdir

directory for FSG files

-fsgext

extension for FSG files (including leading dot)

-fsgusealtpron

Add alternate pronunciations to FSG

-fsgusefiller

Insert filler words at each state.

-fwdflat

Run forward flat-lexicon search over word lattice (2nd pass)

-fwdflatbeam

Beam width applied to every frame in second-pass flat search

-fwdflatefwid

Minimum number of end frames for a word to be searched in fwdflat search

-fwdflatlw

Language model probability weight for flat lexicon (2nd pass) decoding

-fwdflatsfwin

Window of frames in lattice to search for successor words in fwdflat search

-fwdflatwbeam

Beam width applied to word exits in second-pass flat search

-fwdtree

Run forward lexicon-tree search (1st pass)

-hmm containing acoustic model files.

-hyp output file name

-hypseg

output with segmentation file name

-input\_endian

Endianness of input data, big or little, ignored if NIST or MS Wav

-jsgf grammar file

-keyphrase to spot

-kws file with keyphrases to spot, one per line

-kws\_delay Delay to wait for best detection score

-kws\_plp Phone loop probability for keyword spotting

-kws\_threshold Threshold for p(hyp)/p(alternatives) ratio

-latsize

Initial backpointer table size

-lda containing transformation matrix to be applied to features (single-stream features

only)

-ldadim

Dimensionality of output of feature transformation (0 to use entire matrix)

-lifter

Length of sin-curve for liftering, or 0 for no liftering.

-lm

Trigram language model input file

-lmctl

A set of language model

The -hmm and -dict arguments are always required. Either -lm or -fsg is required,

depending on whether you are using a statistical language model or a finite-state grammar.

Mid Term Paper Review Skeleton

* **Speech Recognition**
  + **Introduction**
    - Low resource language
    - Shared cultural history (sanskrit) with hindi
    - State of modern bangla NLP
    - Terminology rundown (mention HMMs, Sphinx and phonemes)
  + **Data**
    - A Contrastive Analysis of English and Bangla Phonemics
    - Development of Annotated Bangla Speech Corpora (CRBLP)
    - [Bangla Speech Corpus for Continuous A](https://cse.iitkgp.ac.in/~pabitra/paper/ococosda11.pdf)SR (SHRUTI)
  + **Techniques**
    - Bangla Speech Recognition using LPC and ANN
    - Connected Bangla Speech Recognition using ANN
    - Bangla Short Speech Commands Recognition Using CNN
    - Continuous Bengali Speech Recognition Based On DNN
  + **Features**
    - Local Features or MFCC for MLNN-based Bangla Speech
    - Noise Robust End-to-End Speech Recognition for Bangla
    - Bangla Word Recognition using Different Acoustic Features
    - Bangla Isolated Speech Recognition
    - G2P: Grapheme to Phoneme for Bangla
* Semantics
  + Approaches for Bangla Question Detection
  + Word to Sentence Level Emotion Tagging for Bengali Blogs
  + A Bangla Semantic Parser Using Context-Free Grammar
  + Semantic Textual Similarity in Bengali Text
  + Sentiment Analysis on Bangla using Deep Recurrent Networks
  + Sentiment Analysis on Bangla Micro Blogs
* Machine Translation
  + ...assignment paper
* Generation
  + Writing like Tagore