

## References

- [1] "Bbc.co.uk, 'BBC - Languages - Languages - Languages of the world - Interesting facts about languages.'" [Online]. Available: <http://www.bbc.co.uk/languages/guide/languages.shtml>. [Accessed: 14-Nov-2015].
- [2] A. Wasala and K. Gamage, "Research Report on Phonetics and Phonology of Sinhala," University of Colombo School of Computing, Colombo.
- [3] "CMU Sphinx."
- [4] T. Nadungodage and R. Weerasinghe, "Continuous Sinhala speech recognizer," in *Conference on Human Language Technology for Development, Alexandria, Egypt*, 2011, pp. 2–5.
- [5] S. Furui, "AUTOMATIC SPEECH RECOGNITION AND ITS APPLICATION TO INFORMATION EXTRACTION." Tokyo institute of Technology.
- [6] "Neural Networks." [Online]. Available: [https://www.doc.ic.ac.uk/~nd/surprise\\_96/journal/vol4/cs11/report.html](https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html). [Accessed: 28-Jan-2016].
- [7] T. Nadungodage, V. Welgama, and R. Weerasinghe, "Developing a Speech Corpus for Sinhala Speech Recognition."
- [8] C. Richey, "Speech Corpora." Apr-2000.
- [9] S. Chen, D. Beeferman, and R. Rosenfeld, "EVALUATION METRICS FOR LANGUAGE MODELS," Carnegie Mellon University, Pittsburgh, PA 15213.
- [10] N. a. C. Sandasarani, "Sinhala Speech Recognition," *Int. J. Eng. Res. Technol.*, vol. Vol. 4, no. Issue 10, pp. 391–394, Oct. 2015.
- [11] S. Molau, M. Pitz, R. Schlüter, and H. Ney, "COMPUTING MEL-FREQUENCY CEPSTRAL COEFFICIENTS ON THE POWER SPECTRUM," RWTH Aachen – University of Technology, 52056 Aachen, Germany.
- [12] C. A. Ratanamahatana and E. Keogh, "Everything you know about Dynamic Time Warping is Wrong," University of California, Riverside, CA 92521.
- [13] L. A. D. S. A. Molligoda and P. G. Wijayarathna, "APPLICABILITY OF HIDDEN MARKOV MODEL APPROACH FOR SINHALA SPEECH RECOGNITION – A SYSTEMATIC REVIEW," University of Kelaniya.

- [14] Z. Ghahramani, "An introduction to Hidden Markov Model and Bayesian Networks," *Int. J. Pattern Recognit. Artif. Intell.*
- [15] P. Dymarski, *HIDDEN MARKOV MODELS, THEORY AND APPLICATIONS*, First. India: InTech, 2011.
- [16] D. Reynolds, "Gaussian Mixture Models." MIT Lincoln Laboratory.
- [17] M. K. H. Gunasekara and R. G. N. Meegama, "Real time Translation of Discrete Sinhala Speech to Unicode Text," University of Sri Jayewardenepura.
- [18] T. Gulzar, A. Singh, and S. Sharma, "Comparative Analysis of LPCC, MFCC and BFCC for the Recognition of Hindi Words using Artificial Neural Networks," *Int. J. Comput. Appl.*, vol. Volume 101–No.12, p. 6, Sep. 2014.
- [19] F. Honig, G. Stemmer, C. Hacker, and F. Brugnara, "Revising Perceptual Linear Prediction (PLP)," Universität Erlangen-Nürnberg, Germany.
- [20] U. Shrawankar, "TECHNIQUES FOR FEATURE EXTRACTION IN SPEECH RECOGNITION SYSTEM : A COMPARATIVE STUDY," SGB Amravati University, Amravati.
- [21] U. Shrawankar and V. Thakare, "Feature Extraction for a Speech Recognition System in Noisy Environment: A Study," 2010, pp. 358–361.
- [22] "Open-Source Large Vocabulary CSR Engine Julius." [Online]. Available: [http://julius.osdn.jp/en\\_index.php](http://julius.osdn.jp/en_index.php). [Accessed: 03-Feb-2016].
- [23] M. A. Anusuya and S. K. Katti, "Speech Recognition by Machine: A Review," *Int. J. Comput. Sci. Inf. Secur.*, vol. Vol. 6, p. 25, 2009.
- [24] K. Lee, *Context-independent phonetic hidden Markov models for speaker-independent continuous speech recognition*, vol. vol. 38, 4 vols. 1990.
- [25] K. Lee, "Automatic speech recognition." Boston, Mass: Kluwer, 1989.
- [26] "GenTXWarper - Mining gene expression time series."
- [27] "The Google app," *The Google app*. [Online]. Available: <https://www.google.com/search/about/>. [Accessed: 04-Mar-2016].
- [28] "iOS - Siri," *Apple*. [Online]. Available: <http://www.apple.com/ios/siri/>. [Accessed: 04-Mar-2016].

- [29] "What is Cortana? - Windows Help," *windows.microsoft.com*. [Online]. Available: <http://windows.microsoft.com/en-us/windows-10/getstarted-what-is-cortana>. [Accessed: 04-Mar-2016].
- [30] "Language Technology Research Laboratory." [Online]. Available: <http://www.ucsc.cmb.ac.lk/ltr/?page=downloads&lang=en&style=default>. [Accessed: 04-Mar-2016].
- [31] "cmusphinx/pocketsphinx," *GitHub*. [Online]. Available: <https://github.com/cmusphinx/pocketsphinx>. [Accessed: 04-Mar-2016].
- [32] "Training Acoustic Model For CMUSphinx [CMUSphinx Wiki]." [Online]. Available: <http://cmusphinx.sourceforge.net/wiki/tutorialam>. [Accessed: 04-Mar-2016].
- [33] "Audacity®." .
- [34] "Cygwin." [Online]. Available: <https://www.cygwin.com/>. [Accessed: 04-Mar-2016].



## **Appendix A**

1. SMS phrases for Greetings
2. SMS phrases for asking Questions
3. SMS phrases related to Relationships
4. SMS phrases related to Students & Exams
5. SMS phrases to mention When & Where
6. SMS phrases for Invitations
7. SMS phrases for appreciation
8. SMS phrases for Emergency
9. SMS phrases to express feelings

# Appendix B

## Frequently Used SMS Phrases

This is a survey to identify what are the most common Sinhala phrases that can be used in SMS. There are 09 categories stated here. For each category sample phrases were given and you are free to suggest any sinhala phrase that matches to the category. The phrases you suggest can be written in singlish or sinhala Unicode by separating each using comma in the given space. You can use Google Transliterator tool to type sinhala Unicode characters using this link (<http://www.google.com/inputtools/try/>) and copy and past the text to the answer box. (Responses to this survey is purely for academic research purposes only)

**What are the frequently used SMS phrases for Greetings?**

Examples: සාදරයෙන් පිළිගන්නවා, සාදරයෙන්, සුභ උදෑසනක්, සුභ දවසක්, සුභ සැන්දෑවක්, සුභ රාත්‍රියක්, සුභ දවසක්

**What are the frequently used SMS phrases to ask Questions?**

Examples: වයස කීයද, වයස කවිද, කුස ඒකාලයේ රැස්විටිම හරක්ක, වයස කොපොයිද හැරී, වයස කීයක් ගත්තද, වයස කොපොද දන්නේ?

**What are the frequently used SMS phrases related to Relationships?**

Examples: වයස විට හදවතෙයි, වීථි පිටි හදවතයි, පිටි වයසට ගොඩක් හදවතයි, වාක් එන්න හරකද

# Appendix C

Frequently (Responses)		
Timestamp	D	E
1	What are the frequently used SMS phrases to ask Questions?	What are the frequently used SMS phrases related to Relationships? What are the frequently used SMS phrases related to Students & Exams?
2	What are the frequently used SMS phrases to ask Questions?	What are the frequently used SMS phrases related to Relationships? What are the frequently used SMS phrases related to Students & Exams?
3	What are the frequently used SMS phrases to ask Questions?	What are the frequently used SMS phrases related to Relationships? What are the frequently used SMS phrases related to Students & Exams?
4	What are the frequently used SMS phrases to ask Questions?	What are the frequently used SMS phrases related to Relationships? What are the frequently used SMS phrases related to Students & Exams?
5	What are the frequently used SMS phrases to ask Questions?	What are the frequently used SMS phrases related to Relationships? What are the frequently used SMS phrases related to Students & Exams?
6	y?	(
7	where are u?	(
8	whats up ???	love you
9	hi	love you
10	What are the frequently used SMS phrases to ask Questions?	What are the frequently used SMS phrases related to Relationships? What are the frequently used SMS phrases related to Students & Exams?



## Appendix D

\data\  
ngram 1=578  
ngram 2=3540  
ngram 3=407

\1-grams:

-5.2121 0 -0.1680

-5.2121 1 -0.1680

-5.2121 2 -0.1680

-5.2121 3 -0.1680

-5.2121 4 -0.1680

-5.2121 5 -0.1680

-5.2121 6 -0.1680

-5.2121 7 -0.1680

-5.2121 8 -0.1680

-5.2121 9 -0.1680

-0.5786 </s> 0.0000

-99.0000 <s> -1.0147

-3.6439 අංකය -0.4351

-3.3992 අභහරුවාදා -0.2699

-4.2170 අඩන්න 0.0000

-3.4412 අතරමං -0.6011

-3.2078 අත්සන් -0.7666

-4.1707 අතුරුපස -0.1328

-3.3488 අඟේ -0.2976

-1.9804 අද -0.3949

-4.2170 අදනම් 0.0000

-5.2121 අදවත් -0.3009

- 2.6619 අධ්‍යාපන -1.3828
- 3.5589 අන්තිම -0.3010
- 4.1707 අනිද්දා -0.1657
- 4.5131 අනිවාර්යෙන්ම -0.1680
- 3.2436 අනුමත -0.6554
- 3.3036 අනේ -0.4151
- 4.5131 අප්පෙට් 0.0000
- 1.9793 අපි -0.1885
- 2.7921 අපිට -0.3704
- 1.9561 අපේ -0.5345
- 2.8055 අමතක -0.8056
- 4.2170 අමතකම 0.0000
- 4.3670 අමතන්න -0.6451



## Appendix E

ආශ්‍රිතවත්	AAZ Y U B O O V N
සුභ	SHZ U B
උදාසනක්	U DHZ AAE SHZ N K
රාත්‍රියක්	R AA THZ RA R I Y K
දවාලක්	DHZ V AA L K
සැන්දෑවක්	SHZ AE N DHZ AAE V K
කොහොමද	K O H O M DHZ
බුදුසරණයි	B U DHZ U SHZ R N Y I
දවසක්	DHZ V SHZ K
වේවා	V EE V AA
දෙව්	DHZ E V I

## Appendix F

අ	a
ආ	aa
ඇ	ae
ඈ	aw
ඉ	i
ඊ	ii
උ	u
ඌ	uu
ඍ	ri
ඎ	rii
ඏ	ilu
ඐ	iluu
එ	e
ඒ	ee
ඓ	ai
ඔ	o
ඕ	oo
ඖ	au
඗	k
඘	k
඙	g
ක	g
ඛ	q
ඛ	ngz
ඞ	c
ඟ	c
ච	j
ඡ	j
ඣ	cnz



ꠄ  
ꠅ  
꠆  
ꠇ  
ꠈ  
ꠉ  
ꠊ  
ꠋ  
ꠌ  
ꠍ  
ꠎ  
ꠏ  
ꠐ  
ꠑ  
ꠒ  
ꠓ  
ꠔ  
ꠕ  
ꠖ  
ꠗ  
ꠘ  
ꠙ  
ꠚ  
ꠛ  
ꠜ  
ꠝ  
ꠞ  
ꠟ  
ꠠ  
ꠡ  
ꠢ  
ꠣ  
ꠤ  
ꠥ  
ꠦ  
ꠧ  
꠨  
꠩  
꠪  
꠫  
꠬  
꠭  
꠮  
꠯  
꠰  
꠱  
꠲  
꠳  
꠴  
꠵  
꠶  
꠷  
꠸  
꠹  
꠺  
꠻  
꠼  
꠽  
꠾  
꠿

jcuz  
njz  
t  
t  
d  
d  
n  
ndz  
thz  
thz  
dhz  
dhz  
n  
ndz  
p  
p  
b  
b  
m  
mbz  
y  
r  
l  
v  
shz  
shz  
shz  
h  
l  
f  
q  
h



## Appendix G

NDZ

AEZ

EEZ

II

OOZ

J

NGZ

IIZ

SIL

+GARBAGE+

+NOISE+

+BREATH+

## Appendix H

<s> SIL

</s> SIL

<sil> SIL

++CHAIRSQUEAK++

+NOISE+

++COUGH++

+NOISE+

## Appendix I

tracks/713-Audio\_Track  
tracks/714-Audio\_Track  
tracks/715-Audio\_Track  
tracks/716-Audio\_Track  
tracks/717-Audio\_Track  
tracks/718-Audio\_Track  
tracks/719-Audio\_Track  
tracks/720-Audio\_Track  
tracks/721-Audio\_Track  
tracks/722-Audio\_Track  
tracks/723-Audio\_Track  
tracks/724-Audio\_Track  
tracks/725-Audio\_Track  
tracks/726-Audio\_Track  
tracks/727-Audio\_Track  
tracks/728-Audio\_Track



## Appendix J

<s> මට තව පැය භාගෙකින් කෝල් කරන්න </s> (1390-Audio\_Track)

<s> මට තව විනාඩි පහක් දෙන්න </s> (1391-Audio\_Track)

<s> 0 </s> (1392-Audio\_Track)

<s> 1 </s> (1393-Audio\_Track)

<s> 2 </s> (1394-Audio\_Track)

<s> 3 </s> (1395-Audio\_Track)

++SINGING++ (noise\_0001)

++MOVEMENT++ (noise\_0002)

++TYPING++(noise\_0005)

## Appendix K

```
package edu.cmu.pocketsphinx.sinhala;
import java.io.File;
import java.util.concurrent.LinkedBlockingQueue;
import android.media.AudioFormat;
import android.media.AudioRecord;
import android.media.MediaRecorder;
import android.os.Bundle;
import android.os.Environment;
import android.util.Log;
import edu.cmu.pocketsphinx.Config;
import edu.cmu.pocketsphinx.Decoder;
import edu.cmu.pocketsphinx.Hypothesis;
import edu.cmu.pocketsphinx.pocketsphinx;
public class RecognizerTask implements Runnable {

    class AudioTask implements Runnable {
        /**
         * Queue on which audio blocks are placed.
         */
        LinkedBlockingQueue<short[]> q;
        AudioRecord rec;
        int block_size;
        boolean done;

        static final int DEFAULT_BLOCK_SIZE = 1024;

        AudioTask() {
            this.init(new LinkedBlockingQueue<short[]>(0), DEFAULT_BLOCK_SIZE);
        }
    }
}
```

```
AudioTask(LinkedBlockingQueue<short[]> q) {  
    this.init(q, DEFAULT_BLOCK_SIZE);  
}
```

```
AudioTask(LinkedBlockingQueue<short[]> q, int block_size) {  
    this.init(q, block_size);  
}
```

```
void init(LinkedBlockingQueue<short[]> q, int block_size) {  
    this.done = false;  
    this.q = q;  
    this.block_size = block_size;  
    //HiepNH - CHANNEL_IN_MONO -> CHANNEL_IN_STEREO  
    this.rec = new AudioRecord(MediaRecorder.AudioSource.DEFAULT, 16000,  
        AudioFormat.CHANNEL_IN_MONO,  
        AudioFormat.ENCODING_PCM_16BIT, 8192);  
}
```

```
public int getBlockSize() {  
    return block_size;  
}
```

```
public void setBlockSize(int block_size) {  
    this.block_size = block_size;  
}
```

```
public LinkedBlockingQueue<short[]> getQueue() {  
    return q;  
}
```



```
public void stop() {
    this.done = true;
}
```

```
public void run() {
    this.rec.startRecording();
    while (!this.done) {
        int nshorts = this.readBlock();
        if (nshorts <= 0)
            break;
    }
    this.rec.stop();
    this.rec.release();
}
```

```
int readBlock() {
    short[] buf = new short[this.block_size];
    int nshorts = this.rec.read(buf, 0, buf.length);
    if (nshorts > 0) {
        Log.d(getClass().getName(), "Posting " + nshorts + " samples to queue");
        this.q.add(buf);
    }
    return nshorts;
}
}
```

```
/**
```

```
 * PocketSphinx native decoder object.
```

```
 */
```

```
Decoder ps;
```

```
/**
```

```

* Audio recording task.
*/
AudioTask audio;
/**
* Thread associated with recording task.
*/
Thread audio_thread;
/**
* Queue of audio buffers.
*/
LinkedBlockingQueue<short[]> audioq;
/**
* Listener for recognition results.
*/
RecognitionListener rl;
boolean use_partials;

enum State {
    IDLE, LISTENING
};
enum Event {
    NONE, START, STOP, SHUTDOWN
};

Event mailbox;

public RecognitionListener getRecognitionListener() {
    return rl;
}

public void setRecognitionListener(RecognitionListener rl) {

```

```

    this.rl = rl;
}

public void setUsePartials(boolean use_partials) {
    this.use_partials = use_partials;
}

public boolean getUsePartials() {
    return this.use_partials;
}

public RecognizerTask(File sphinxDirectory) {
    pocketsphinx.setLogfile(sphinxDirectory.getAbsolutePath()+"/pocketsphinx.log");
    Config c = new Config();
    c.setString("-hmm", sphinxDirectory + "/hmm/sinhala");
    c.setString("-dict", sphinxDirectory + "/dict/test06.dic");
    c.setString("-lm", sphinxDirectory + "/lm/test06.lm");
    c.setString("-rawlogdir", sphinxDirectory.getAbsolutePath()); // Only use it to store
the audio
    c.setFloat("-samprate", 16000.0);
    c.setInt("-maxhmpf", 2000);
    //c.setInt("-maxwfp", 10);
    //c.setInt("-pl_window", 2);
    //c.setBoolean("-backtrace", true);
    //c.setBoolean("-bestpath", false);
    this.ps = new Decoder(c);
    this.audio = null;
    this.audioq = new LinkedBlockingQueue<short[]>(0);
    this.use_partials = true;
    this.mailbox = Event.NONE;
}

public void run() {
    /* Main loop for this thread. */

```



```

boolean done = false;
/* State of the main loop. */
State state = State.IDLE;
/* Previous partial hypothesis. */
String partial_hyp = null;

while (!done) {
    Event todo = Event.NONE;
    synchronized (this.mailbox) {
        todo = this.mailbox;
        if (state == State.IDLE && todo == Event.NONE) {
            try {
                Log.d(getClass().getName(), "waiting");
                this.mailbox.wait();
                todo = this.mailbox;
                Log.d(getClass().getName(), "got" + todo);
            } catch (InterruptedException e) {
                /* Quit main loop. */
                Log.e(getClass().getName(), "Interrupted waiting for mailbox, shutting
down");
                todo = Event.SHUTDOWN;
            }
        }

        this.mailbox = Event.NONE;
    }
    switch (todo) {
    case NONE:
        if (state == State.IDLE)
            Log.e(getClass().getName(), "Received NONE in mailbox when IDLE,
threading error?");

```

```

    break;
case START:
    if (state == State.IDLE) {
        Log.d(getClass().getName(), "START");
        this.audio = new AudioTask(this.audioq, 1024);
        this.audio_thread = new Thread(this.audio);
        this.ps.startUtt();
        this.audio_thread.start();
        state = State.LISTENING;
    }
    else
        Log.e(getClass().getName(), "Received START in mailbox when
LISTENING");
    break;
case STOP:
    if (state == State.IDLE)
        Log.e(getClass().getName(), "Received STOP in mailbox when IDLE");
    else {
        Log.d(getClass().getName(), "STOP");
        assert this.audio != null;
        this.audio.stop();
        try {
            this.audio_thread.join();
        }
        catch (InterruptedException e) {
            Log.e(getClass().getName(), "Interrupted waiting for audio thread, shutting
down");
            done = true;
        }
        short[] buf;
        while ((buf = this.audioq.poll()) != null) {

```

```

        Log.d(getClass().getName(), "Reading " + buf.length + " samples from
queue");
        this.ps.processRaw(buf, buf.length, false, false);
    }
    this.ps.endUtt();
    this.audio = null;
    this.audio_thread = null;
    Hypothesis hyp = this.ps.getHyp();
    if (this.rl != null) {
        if (hyp == null) {
            Log.d(getClass().getName(), "Recognition failure");
            this.rl.onError(-1);
        }
        else {
            Bundle b = new Bundle();
            Log.d(getClass().getName(), "Final hypothesis: " + hyp.getHypstr());
            b.putString("hyp", hyp.getHypstr());
            this.rl.onResults(b);
        }
    }
    state = State.IDLE;
}
break;
case SHUTDOWN:
    Log.d(getClass().getName(), "SHUTDOWN");
    if (this.audio != null) {
        this.audio.stop();
        assert this.audio_thread != null;
        try {
            this.audio_thread.join();
        }
    }
}

```





```

    catch (InterruptedException e) {
    }
}
this.ps.endUtt();
this.audio = null;
this.audio_thread = null;
state = State.IDLE;
done = true;
break;
}
if (state == State.LISTENING) {
    assert this.audio != null;
    try {
        short[] buf = this.audioq.take();
        Log.d(getClass().getName(), "Reading " + buf.length + " samples from queue");
        this.ps.processRaw(buf, buf.length, false, false);
        Hypothesis hyp = this.ps.getHyp();
        if (hyp != null) {
            String hypstr = hyp.getHypstr();
            if (hypstr != partial_hyp) {
                Log.d(getClass().getName(), "Hypothesis: " + hyp.getHypstr());
                if (this.rl != null && hyp != null) {
                    Bundle b = new Bundle();
                    b.putString("hyp", hyp.getHypstr());
                    this.rl.onPartialResults(b);
                }
            }
            partial_hyp = hypstr;
        }
    } catch (InterruptedException e) {
        Log.d(getClass().getName(), "Interrupted in audioq.take");
    }
}

```

