

Powerpoint Controller using Speech Recognition

Christina^{#1}, Rosalina^{#2}, R.B Wahyu^{#3}, Rusdianto Roestam^{#4}

[#] Faculty of Computing, President University,
Jl. Ki Hajar Dewantara, Cikarang Baru – Bekasi 17550

¹christina.thamim@yahoo.com

²rosalina@president.ac.id

³rbw0101@gmail.com

⁴rroestam@gmail.com

Abstract — During presentation, it is hard to maintain the slide because we need to stand in front of the room and often not able to touch the computer. The presenters need to take attention at both their voice, and body language such eye contact, facial expression, posture, gesture, and body orientation. Microsoft PowerPoint is a simple but very useful tool to create digital presentation. Even though it is simple to use, but this application required the presenter to take control while using it, such as to start the slide show or moving it to the next slide. The purpose of this research is to minimize physical contact between user and the computer during the presentation by controlling the move of the slide using voice. This research will implement the Hidden Markov Model algorithm and Sphinx-4 library. As a result of this research, the application gives benefit for the user who wants to made their presentation become more effective. The user can use this application from distant, using microphone or other input voice that compatible with their computer.

Keywords— Hidden Markov algorithm, Sphinx-4 library, powerpoint controller, speech recognition.

I. INTRODUCTION

Nowadays, people use the digital presentation to present their idea. Digital presentation makes the audience become more interested, since the speaker can add some picture or video and event design that related to their topic. Microsoft PowerPoint is a simple but very useful tool to create the digital presentation. Even though it is simple to use, but this application required the presenter to take control while using it, such as to start the slide show or moving it to the next slide. The traditional ways to take control this application is by pressing the keyboard shortcut while running the application.

Normally, to interact with the computer user had to give some input message to their computer such as typing the keyboard or using the mouse and get the output on the monitor screen. While for the wireless presenters, they interact with the computer using remote controller, they are limited with the battery signal, and having too many buttons

on a wireless presenter can be frustrating. However, for the users who are not be able to use the method above, they could use voice as an input and output from the computer. The method of using the voice as the output of the computer called as text-to-speech (TTS) while the opposite of this method is called as speech-to-text (STT). Speech-to-text or often called as speech recognition is perform when user replacing the input with their voice.

Many algorithm have been used to improve the speech recognition system, such as Hidden Markov Model (HMM), Dynamic Time Warping (DTW) and Neural Network (NN). In comparison to the other algorithm, the HMM algorithm which tend to be more popular will be used as the basic for speech recognition in this research. However, the environment and some other condition such as the size of vocabulary and speaker dependency give many contributions with the performance of the speech recognition.

The main objective of this research is to develop an application that implements the Hidden Markov Model algorithm in order to recognize the speaker's voice and give command to Microsoft PowerPoint. The application purpose is to help people to control the presentation slide using their own voice.

The application will do the recognition using speech recognition library that implements the Hidden Markov Model in the system. The application will have some feature such as opening Microsoft PowerPoint application and adjustment feature. The adjustment feature is provided to give time for the microphone and application to adjust with the environment. The speaker dependency will not applied in this application, therefore this application will not cover feature for training the application by user's voice. The command that could be given in order to controlling the PowerPoint by the user are starting the presentation show, starting the presentation show from current slide, exiting the presentation show, perform option to choose slide, change slide to the next or previous slide, performing right mouse click, performing the arrow (up, down, left, right), and the

enter button.

II. LITERATURE STUDY

A. Speech Recognition

Speech recognition is the process of transforming a spoken word into text. Speech Recognition involves capturing the user's utterance, digitizing utterance into a digital signal then converting them into basic units of utterance and contextually analyzing the words to ensure correct spelling for words that sound alike [7]. When user speaks into the microphone, the system catch user's voice in the form of analog acoustic signal and generates the electrical impulses. Then the sound card will convert the acoustic signal into the digital signal and the speech recognition engine will process the digital signal into phonemes, where phonemes will be creating the words.

There is a differentiation between the speech recognition and the speaker recognition. Speech recognition is the process of understanding for what being said, while the speaker recognition is the process of recognizing who is speaking. In other word, speaker recognition is the process to determine the person that giving input voice. The main motivation of developing speech recognition is to improve communication between human and computer by creating the artificial model of human verbal communication [1].

There is some techniques approach used to produce the best result of speech recognition:

- Template based approaches matching
In order to find the best result, the input speech is compared against a set of pre-recorded templates. Using a perfectly accurate word model gives more advantages than the other technique. However, the model is fixed and inflexible where the environment background will be another issue of its disadvantages.
- Knowledge based approaches.
Hand coded of an expert knowledge about variations of speech, however it judged as impractical due to its difficulties in obtaining and used though it has the advantage of explicit modeling variations in speech.
- Statistical based approaches.
The speech is modelled statistically, using automatic and statistical learning procedure, which represents the current state of the art. This is the type of the Hidden Markov Models, or HMM. The main disadvantage is that the system must take priori-modelling assumptions of which are liable to be inaccurate, handicapping the system performance.
- Learning based approaches.
Machine learning methods could be introduced as the solution of HMM's disadvantages, for example the neural networks and genetic algorithm. The system can learn automatically through emulations or evolutionary process.
- The artificial intelligence approach

The recognition procedure attempt to apply the artificial intelligence in making the decision. The expert system is used widely in this approach.

B. Hidden Markov Model [6,8,10,12,13]

A Hidden Markov Model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states which developed by L. E. Baum and coworkers [5]. The HMM are acted as a more controlled approaches in the recognition/classification based systems. The HMM are the state of art techniques to represent various speech units characteristics within the model parameters. The HMMs are like Markov Chains in which the output symbols and the transitions are probabilistic. The HMMs represent speech as a sequence of observation vectors derived from a probabilistic function of a first-order Markov chain. In speech recognition, the HMMs would output a sequence of n -dimensional real-valued vectors. In each state statistical distribution gives likelihood for each observed vector i.e. for each word will have a different output distribution. The decoding of the speech means computing the most likely word from an unknown utterance.

The HMM are generative models the state space of the hidden variables is discrete, while the observations themselves can either be discrete generated from a categorical distribution or continuous generated from a Gaussian distribution. The HMM has two parameters for modeling. The joint distribution of observations and hidden states means the prior distribution of hidden states called transition probabilities and conditional distribution of observations given states called emission probabilities. In the standard HMM, the algorithm implicitly assumes a uniform prior distribution over the transition probabilities. The transition probabilities are controlled by the hidden states.

The Left-Right model also known as Bakis model, satisfies the property that as time increases the state index increases or stays the same i.e. the states proceed from left to right. It can easily model signals whose properties change over time in a successive manner i.e. speech. The fundamental property of all the left-right HMMs is that the state transition coefficients have the property:

$$a_{i,j} = 0, j < i \quad (1)$$

that is no transitions are allowed to states whose indices are lower than the current state. The initial state probabilities have the property

$$\pi_i = 0, i \neq 1 \text{ otherwise } 1 \quad (2)$$

since the state sequence must begin in state 1 and end in state N . In Left-Right model additional constraints are placed on the state transition coefficients in the form of

$$a_{ij} = 0, j > i + \Delta \quad (3)$$

Where $\Delta = 2$ i.e. no jumps of more than 2 states are allowed. The last state in left-right model is specified by state transition coefficients are

$$a_{NN} = 1, a_{Ni} = 0, i < N \tag{4}$$

The constraints are useful to make sure that large changes in state indices do not occur. There is no effect on the re-estimation procedure due to constraints on left-right model because any HMM parameter set to zero throughout the re-estimation procedure.

The goal of HMM parameter estimation is to maximize the likelihood of the data under the given parameter setting [3].

Figure 1 is an example of search graph of speech recognition in Hidden Markov Model Algorithm, but the real search graph will be more complex than this example. This example give illustration of construct word & sentence-level HMMs from the phoneme-level units. Each word decomposed into pronunciation, for example, “ONE” with pronunciation “W AX N”, and “TWO” with pronunciation “T OO”, and each of the pronunciation decomposed into the phoneme-level unit.

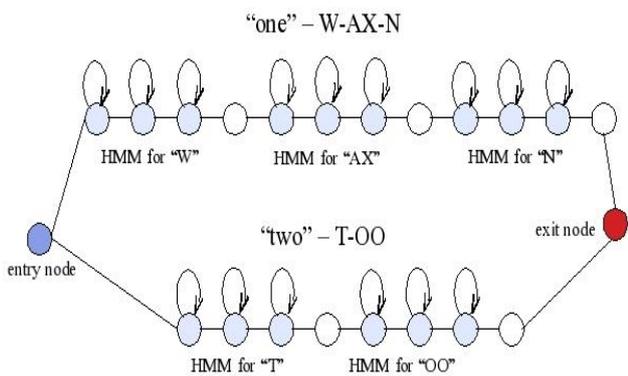


Figure 1. Illustration of Search Graph HMM [5]

Hidden Markov Model is a variant of a finite state machine. However, unlike the finite state machines which emits a deterministic symbol in a given state and deterministically do the transition to another state, the Hidden Markov Models do both of the transition and emit under a probabilistic model.

In the finite state machine, a string of symbols can be given and it can be easily determined whether the finite state machine could have generated the string in the first place and if it could, what the sequence of state transitions it undertook were. With a hidden Markov model, the determination process for generating string is replaced with a probability that the HMM generated the string and the sequence of state transitions is replaced with nothing or “hidden”.

C. CMU Sphinx

The CMU Sphinx is an open-source speech recognition system created by the Carnegie Mellon University (CMU) Sphinx. The system created by this group are Sphinx, Sphinx-2, Sphinx-3, Sphinx-4 and Pocket Sphinx.

Sphinx-4 is one of speech recognition software written in Java language. It was created by a collaboration of Carnegie Mellon University, Sun Microsystems Laboratories, Mitsubishi Electric Research Labs (MERL), and Hewlett Packard (HP), with contributions from the University of California at Santa Cruz (UCSC) and the Massachusetts Institute of Technology (MIT) [2].

In the Sphinx-4 architecture, there is a front-end module which will parameterize the speech signal, and communicate the derive feature to decoding block [5]. As described in the Figure 2, the decoding block has three components: search manager, linguist, and acoustic scorer. The component of the knowledge base module is interact with the linguist component at decoding block.

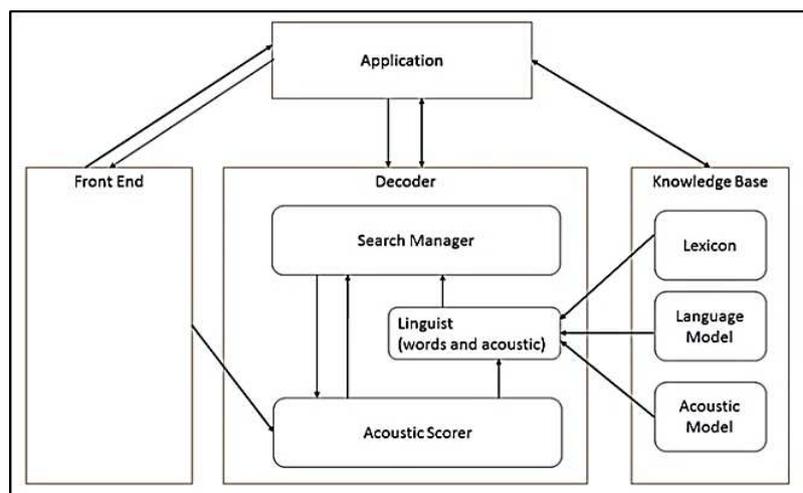


Figure 2. Sphinx-4 System Architecture [5]

The front-end module consist of several communicating blocks as described in Figure 2. Each of these blocks has its own input and output which linked to its predecessor and successor. There is two type of input data that need to be interpreted by the block: the speech data or a control signal. Control signal is an indication of beginning or end of a speech. The output of any of the blocks can be tapped, so it is possible for the actual input of the system is not in the first block, but in any of intermediate blocks. The current implementation also allowed to run the system using not only speech signals, but also spectra, and cepstra.

The endpoint detects the start and/or end of speech from the incoming audio, then sending only the speech segment to the decoder. In this case, the decoder will not waste time processing non-speech segments. There is three modes of operation can be performed: fully endpointed where system do the endpointing automatically of both beginning and ending endpoints, click to talk where the user indicates the beginning but let the system to determine the ending, and push to talk where the user is the one who indicate beginning and ending of the speech segment.

The decoder module performs the actual recognition. This module consist of three blocks: the search manager, the linguist and the acoustic scorer.

The search manager construct and search a tree of possibilities to get the best hypothesis. The search manager make use of a token tree which consist of a set of token that contains information about the search and complete history of all active paths in the search. Each of token contains the overall acoustic and language scorer at the path, a SentenceHMM reference, identification of an input feature frame, and a reference to the previous token. Using the reference of SentenceHMM, the search manager can fully categorize a token to its sentence, context-dependent phonetic unit, pronunciation, word, and grammar state. The information used to create the search tree is references from the linguist.

New implementation of Sphinx-4 can easily be created that provide alternative methods of storing and pruning the active list. The active list is a set of active tokens. This list represents the tips of the active search branches. The Garbage collection will automatically reclaims the unused tokens so it will simplifying the implementation of the pruner. The application can use the active list, the one who available as part of the final recognition, to inspect the highest scoring paths.

There is two ways to perform the search through token and the sentence HMM: depth-first and breadth-first. The depth-first search is similar with the conventional stack decoding. Thus, the length of paths from root to the active token can be vary. While, in the breadth-first search, the length are expanded synchronously, this made the paths from the roots equally long.

D. Microsoft PowerPoint

Presenter, the first name of Microsoft PowerPoint was originally design for Macintosh Operating System. This application developed by Dennis Austin and Thomas Rudkin by Forethought, Inc. The idea of the name was given by Robert Gaskins when the Forethought, Inc. bought by Microsoft [3].

PowerPoint consist of individual pages name as slide. Each slide can be collaboration of text, graphic, movies, sound and any other object. PowerPoint application provide the shortcut both keyboard and mouse button to maximize its performance, such as pressing the CTRL+C to copy and CTRL+V to paste some objects. The shortcut button can also be used while running the slide presentation. The Table I will describe some of the PowerPoint shortcut, especially to run the presentation in the Microsoft PowerPoint 2007, Microsoft PowerPoint 2010, and Microsoft PowerPoint 2013 [9].

TABLE I.

SHORTCUT OF RUN THE PRESENTATION IN MICROSOFT POWERPOINT

Function	Shortcut
Start the presentation from beginning	F5
Start the presentation from current slide	SHIFT + F5
Display a blank black slide	B
Display a blank white slide	W
Display the shortcut menu	Right Mouse Click, SHIFT + F10
Performing next slide	N, ENTER, RIGHT ARROW, DOWN ARROW, or SPACEBAR
Performing previous slide	P, PAGE UP, LEFT ARROW, UP ARROW, or BACKSPACE
Close the presentation	ESC

The PowerPoint Controller which is developed in this research has differences with the other PowerPoint controller, such as open PowerPoint application from application, adjusting microphone and system tray menu. The voice command could be given to the application such as start slide show from current slide, start slide show from the beginning, opening all slide.

III. SYSTEM ANALYSIS

The system is divided into two main parts, which is the speech recognition and controlling PowerPoint. These two classes are the main part of the system. Besides those two

classes, there are also classes that will handle both of the training and the connector. The connector is the bridge class between the result of input voice class and controlling class. There will be a grammar data consist of several words that will be used to give limitation of word that can be recognized by the input voice class.

There are five classes in the PowerPoint Controller application such as LegnaController Class, InputVoice Class, ShowText Class, ChangeText Class and DoCommand Class. LegnaController Class contains the main class of the application. There is also two other file, an xml type for configuration manager and a grammar file provide as known word can be recognize by system.

LegnaController Class contains the graphical user interface of the application menu that can be chosen by the user. The menu is provided in button mode. Another graphical user interface named as adjustment application is provided for user to adjust the application before they could using it.

Feature for opening PowerPoint, stop and exit handled by the LegnaController, while in the starting application, LegnaController class will call for the recognition class named as InputVoice Class. The InputVoice Class can be called either for starting the application, or adjustment the application. If the previous called was from the ShowPanel Class, the InputVoice Class will sent the recognition result to the ShowText Class, while if the previous called was from the LegnaController Class, the InputVoice Class will sent the result to the ChangeText Class. The ChangeText

Class will do the controlling for the Microsoft PowerPoint by calling the DoCommand Class.

The Use-Case Diagram for the Legna PowerPoint Controller is described in Figure 3 (Part 1) and Figure 4 (Part 2).

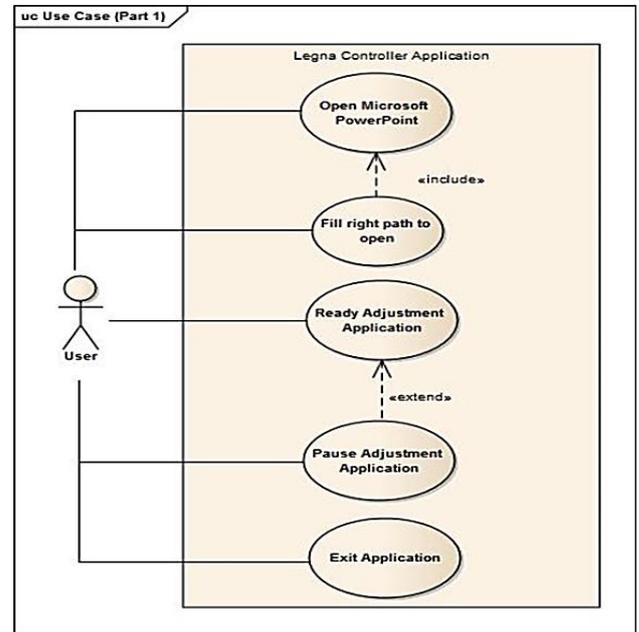


Figure 3. Use-Case Diagram of Legna PowerPoint Controller (Part 1)

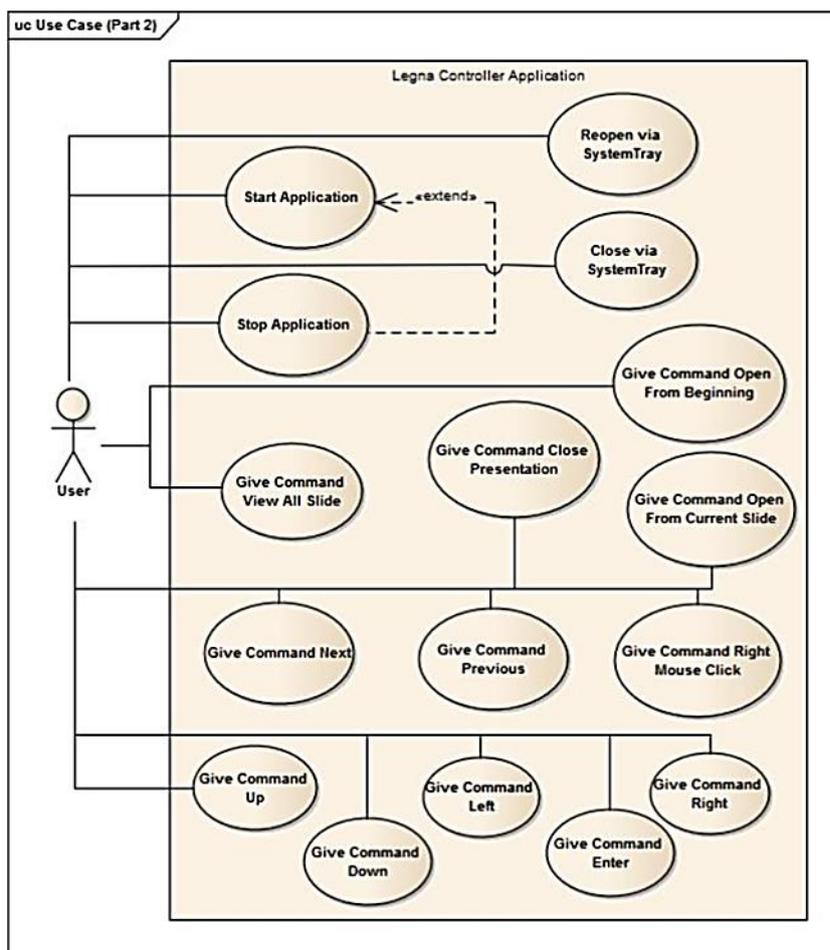


Figure 4. Use-Case Diagram of Legna PowerPoint Controller (Part 2)

There are one actor and twenty-one use-cases in the application. The actor is the user who using the Legna PowerPoint Controller Application. Due to the Use-Case Diagram above, the process that occurs in the applications are:

- 1) Open Microsoft PowerPoint and Fill Right Path to Open

The user can open Microsoft PowerPoint Application using Legna Controller. In some cases, the installation folder of Microsoft PowerPoint in the user's computer is different with the normal path. Therefore, the application will ask the user to provide the right folder path.

- 2) Ready Adjustment and Pause Adjustment

The user will be suggested to use the adjust option before using the application. Adjustment window will be shown when the user starts the application. The user can give some input voice and get feedback to check whether their pronunciation already match with the system. While doing the adjustment, user can choose to pause the adjustment activity by using the pause button.

The input voice can be given to this adjustment is between these words: Open, Close, Start Here, All Slide, Next, Previous, Mouse, Left, Right, Down, Up, and Enter. The command given is same with starting the application, but in adjustment menu, the application will not doing controlling and only take responsibility to give result of the recognition word. While in the starting menu, the system take control of PowerPoint Application. The user can exit the adjustment window by manually selecting the close button of adjustment window.

- 3) Start Application, System Tray, and Give Command

When users select the start button, the application will be added to the system tray and running in the background system. User can reopen the application menu or directly exit the application window by choosing menu at the system tray icon. This activity named as reopen via system tray, and close via system tray.

After pressing the start button, the user can start giving commands for PowerPoint application. All of the give command activity required performing the start application

activity. The input voice can be given to control the applications are:

- Open (Start Slide Show from the beginning)
- Close (Exit the Slide Show)
- Start Here (Start Slide Show from current slide)
- All Slide (Menu for choosing specific slide)
- Next (Performing Next Slide)
- Previous (Performing Previous Slide)
- Mouse (Performing Right Clicked Mouse)
- Left (Performing Left Button Pressed)
- Right (Performing Right Button Pressed)
- Down, (Performing Down Button Pressed)
- Up (Performing Up Button Pressed)
- Enter (Performing Enter Button Pressed)

4) Stop Application

User can stop the application by using the stop button if the user intended to start it again later.

5) Exit Application

User can close the application with select the exit button in the main menu screen and the application will be stopped and closed.

IV. IMPLEMENTATION

A. Implementation of HMM and Sphinx-4

CMU Sphinx-4 is used for the Automatic Speech Recognition part. The sphinx system was chosen because of this research is implemented in Java and because it was open source software and thus it is free from licensing issues. The sphinx recognizer is a Hidden Markov Model based system, during recognition, the audio signal is processed and sets of different features are extracted from it.

The word recognizer define the recognizer component that performs speech recognition. The following code in the recognizer part define the name and class of the recognizer. The Recognizer class has two properties, the 'decoder' and the 'monitors'. In this configuration file, the value of these properties are defined. Figure 5 is the explanation code of word recognizer parts.

```
<component name="recognizer"
type="edu.cmu.sphinx.recognizer.Recognizer">
  <property name="decoder" value="decoder"/>
  <propertylist name="monitors">
    <item>accuracyTracker </item>
    <item>speedTracker </item>
    <item>memoryTracker </item>
  </propertylist>
</component>
```

Figure 5. Code of Word Recognizer parts

The 'decoder' property of the recognizer is set to the component called 'decoder'. The decoder component

property 'searchManager' is set to the component 'searchManager'.

The searchManager class performs a simple breadth-first search through the search graph during the decoding process to find the best path. The simple breadth-first search is suitable for small to medium sized vocabulary decoding.

The linguist of the searchManager is set to the component 'flatLinguist' which will be explained in the linguist configuration parts. Similar with the simple breadth-first search, the flat linguist is suitable for small to medium sized vocabulary decoding. The pruner is set to the 'trivialPruner' which will performs simple absolute beam and relative beam pruning based on the scores of the tokens.

Scoring is one of the most time-consuming step in the decoding process. The scorer of the searchManager is set to the component threadedScorer that can use multiple threads to score the tokens in the active list.

The logMath property is the log math that is used for calculation of scores during the search process. The same log base should be used throughout all components, and therefore there is only one logMath definition in a configuration file that declare in the miscellaneous parts. Figure 6 is the explanation code of logMath component in the miscellaneous parts.

```
<component name="logMath"
type="edu.cmu.sphinx.util.LogMath">
  <property name="logBase" value="1.3001"/>
  <property name="useAddTable" value="true"/>
</component>
```

Figure 6. Code of Miscellaneous parts

The linguist of the searchManager is set to the component 'flatLinguist' which will be explained in the linguist configuration parts. Similar with the simple breadth-first search, the flat linguist is suitable for small to medium sized vocabulary decoding. The pruner is set to the 'trivialPruner' which will performs simple absolute beam and relative beam pruning based on the scores of the tokens.

Scoring is one of the most time-consuming step in the decoding process. The scorer of the searchManager is set to the component 'threadedScorer' that can use multiple threads to score the tokens in the active list. The tokens can be scored independently of each other, therefore using multiple CPUs will definitely speed things up. The 'frontend' property is the front end from which features are obtained. The activeListFactory property of the searchManager is set to the component 'activeList'. The activeList uses a partitioning algorithm to select the top N highest scoring tokens when performing absolute beam pruning. The 'logMath' property in activeList component specifies the logMath used for score calculation, which is the same LogMath used in the searchManager. The property 'absoluteBeamWidth' is set to the value given at the very top

of the configuration file using $\{\text{absoluteBeamWidth}\}$. The same is for $\{\text{relativeBeamWidth}\}$.

The linguist configuration parts, a component inside the searchManager, is the component that generates the search graph using the guidance from the grammar, and knowledge from the dictionary, acoustic model, and language model. This parts also uses the logMath that have been defined in the miscellaneous parts. Figure 7 is the code of linguist configuration parts.

```
<component name="flatLinguist"  
  type="edu.cmu.sphinx.linguist.flat.FlatLinguist">  
  <property name="logMath" value="logMath"/>  
  <property name="grammar" value="jsgfGrammar"/>  
  <property name="acousticModel" value="wsj"/>  
  <property name="wordInsertionProbability"  
    value="\{\text{wordInsertionProbability}\}"/>  
  <property name="languageWeight"  
    value="\{\text{languageWeight}\}"/>  
  <property name="unitManager" value="unitManager"/>  
</component>
```

Figure 7. Code of Linguist Configuration parts

The grammar refer to the component called as 'jsgfGrammar', which is a BNF-style grammar. The

edu.cmu.sphinx.jsapi.JSGFGrammar class will translates JSGF into a form that Sphinx-4 understands. The 'grammarLocation' property is used to specify the location of the resource input.gram. The 'grammarName' property specifies the grammar to use when creating the search graph. The 'logMath' property is the same log math as the other components.

There are two different front ends listed in the configuration file: the frontend and the epFrontEnd. The frontend is good for batch mode decoding without endpointing, while the epFrontEnd is good for live mode decoding with endpointing. The other different between these two front ends is that the live front end has the additional components speechClassifier, speechMarker and nonSpeechDataFilter. These components make up the default endpointer of Sphinx-4.

B. User Interface Development

User Interface Development divided into two sections: Main Menu Interface, and Adjustment Interfaces. When user run the PowerPoint Controller, the application will provide two window, the PowerPoint Controller Main Menu Window and the Adjustment Menu Window.

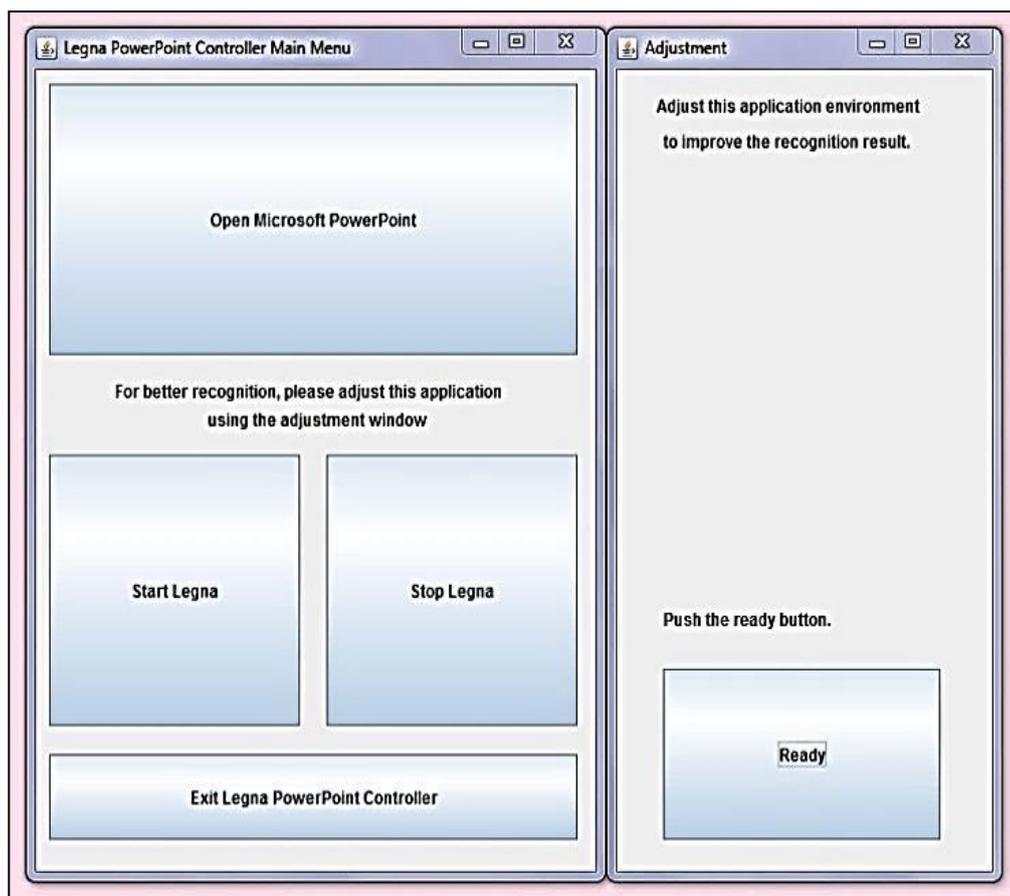


Figure 8. Application Screenshot

In the main menu window, there is a suggestion for user to use the adjustment menu before start using the application. Figure 8 is the print screen of the Main Menu Application Window and Adjustment Window. The two windows are maintained to be displayed side by side.

Main Menu window consist of several menu that can be selected by the user such as open, start, stop and exit. Figure 9 is the print screen of the Legna PowerPoint Controller main menu.

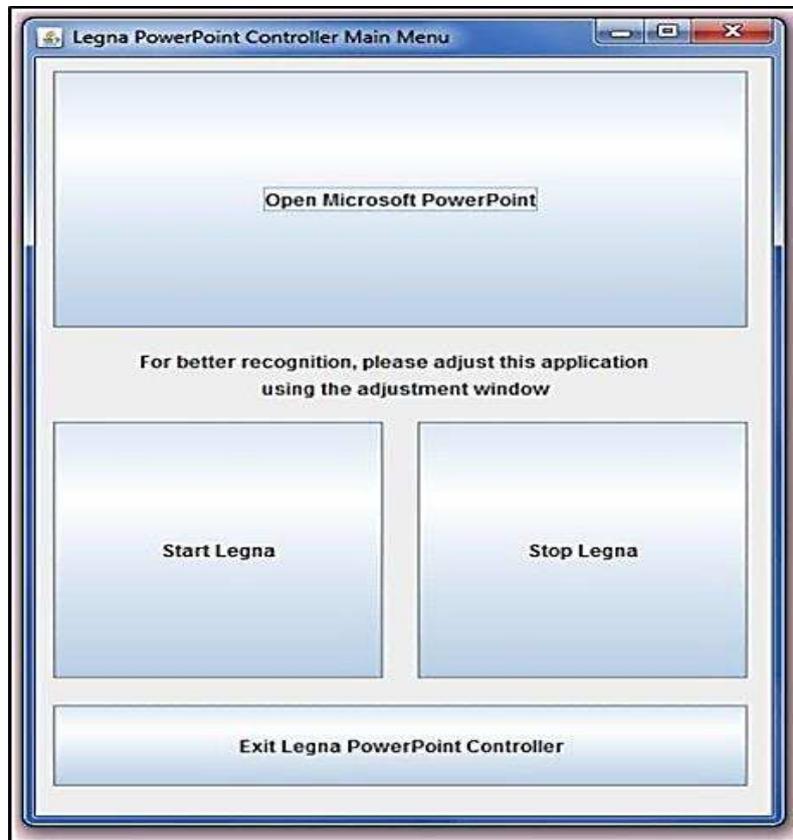


Figure 9. Main Menu Application Screenshot

The open button is provided to use the feature of opening Microsoft PowerPoint from application. When user click the open button, the application will automatically open the Microsoft PowerPoint on user's computer. The

system will search the PowerPoint application using the default path set by the developer. Figure 10 is the print screen of the main menu when user click the open button.

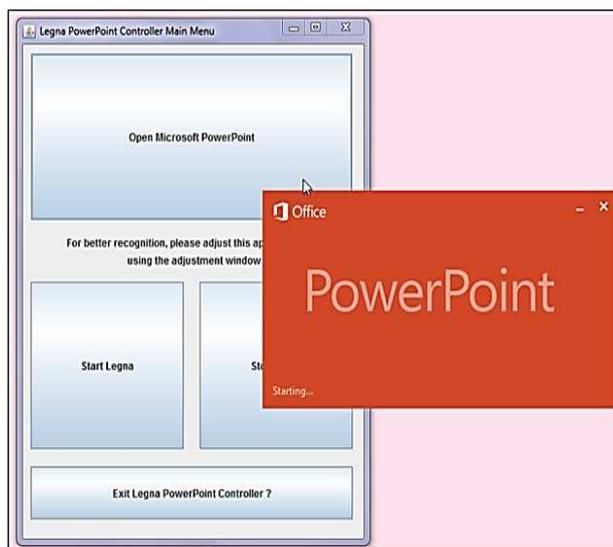


Figure 10. Success Open Microsoft PowerPoint

However, if the application cannot find the Microsoft PowerPoint application, the PowerPoint Controller will provide an input text for user to provide the right path manually, and a button to try again. Figure 11 is the print screen of the main menu window when system cannot find the Microsoft PowerPoint application.

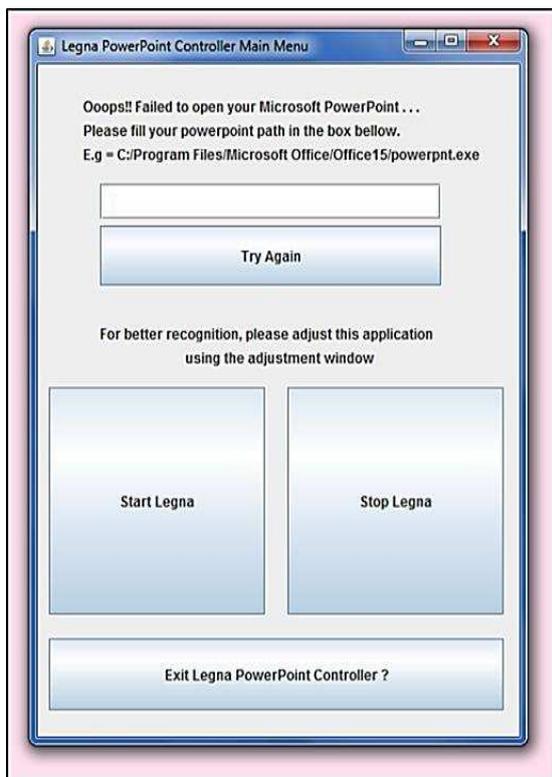


Figure 11. System cannot find PowerPoint application

The user will be asked to fill the path of their PowerPoint application in the path provided. There is an example of right folder path that must be provided by the user. After filling the path for Microsoft PowerPoint Application and pressing the try again button, the system will try to open the Microsoft PowerPoint from the path provided. If the system could find the Microsoft PowerPoint application through that path, the system will launch the Microsoft PowerPoint.

When the user clicks the start button, the system will start the recognition and controlling. The system works in the background of the system, which means the main menu window will be closed and added automatically to the system tray. In order to identify the program in the system tray, the PowerPoint controller uses an icon. In the system tray, the PowerPoint Controller icon will indicate that the application is still running in the background. The popup menu will be displayed when the user clicks the right button mouse on the PowerPoint Controller icon. There are two menu items in the popup menu that can be selected. The first is open and the other is exit. When the user chooses the open option in the popup menu, the main menu window will be reopened. The exit menu item in the popup menu is provided to exit the application directly without reopening the main menu window.

The adjustment menu window will be displayed after the application runs (Figure 12). This window is located beside the main menu. In the adjustment menu, there are two options that are classified as buttons, which can be selected by the user. The menu items are ready and pause. The ready button is provided in order to start the adjustment, while the pause button is provided to pause the adjustment. In the beginning, the pause button is hidden until the ready button is pressed.

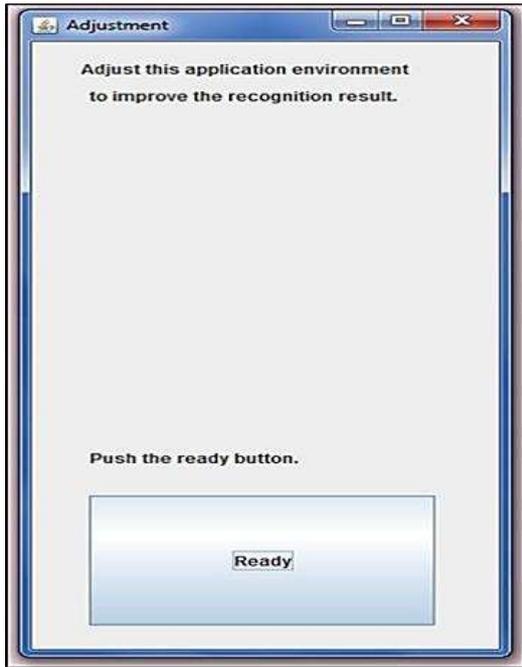


Figure 12. Adjustment Menu's Screenshot

After user press the ready button, the suggestion word will be shown below the label "Please says one of this word in range 5 second". The result word will be shown after user said the command word beside the label "You say:." as the feedback of the recognition. The pause button will be shown while the ready button will be set invisible. Figure 13 is the print screen of adjustment menu when user click the ready button.



Figure 13. Adjustment Menu Window when Ready Button Pressed

The user can pause the adjustment menu by pressing the pause button. When the pause button pressed, the result word and the pause button will be set to invisible, while the ready button will be shown and rename as "Again".

The possible path for the Microsoft PowerPoint listed in a list of string. Table II is the list of possible path for Microsoft PowerPoint 2007-2013 both support for windows 32-bit and the 64-bit.

TABLE II.

PATH FOR MICROSOFT POWERPOINT

Type	Path
PowerPoint 2007	C:/Program Files/Microsoft Office/Office12/powerpnt.exe
PowerPoint 2010	C:/Program Files/Microsoft Office/Office14/powerpnt.exe
PowerPoint 2013	C:/Program Files/Microsoft Office/Office15/powerpnt.exe
PowerPoint 2007 (64 bit)	C:/Program Files (x86)/Microsoft Office/Office12/powerpnt.exe
PowerPoint 2010 (64 bit)	C:/Program Files (x86)/Microsoft Office/Office14/powerpnt.exe
PowerPoint 2013 (64 bit)	C:/Program Files (x86)/Microsoft Office/Office15/powerpnt.exe

In order to start the presentation from current slide, the program will take advantage of the shortcut PowerPoint key, which need to press the SHIFT and F5 button. The other method will also use the shortcut PowerPoint key. The list of command and the shortcut key for each method is described in Table III.

TABLE III.

AVAILABLE COMMAND IN LEGNA POWERPOINT CONTROLLER

Available Command	Method Name	PowerPoint Shortcut
Start presentation from current slide	Start()	SHIFT + F5
Start presentation from first slide	Open()	F5
Close presentation	Close()	ESC
View all slide	All()	Click() + 4x Down() + Enter()
Change to the next slide	Next()	N, Enter, Right Arrow Button

Change to the previous slide	Previous()	P, Left Arrow Button
View shortcut menu	Click()	Right Mouse Click
Perform button up	Up()	Up Arrow Button
Perform button down	Down()	Down Arrow Button
Perform button left	Left()	Left Arrow Button
Perform button right	Right()	Right Arrow Button
Perform button enter	Enter()	Enter Button

Normally, there is no shortcut key for command to view all slide from Microsoft PowerPoint. In PowerPoint Controller, there is a feature for user to view all slide. The flow to view all slide is view shortcut menu, move down four times and press enter. The method *all()* use the flow to provide a feature for user to move the presentation to other slide number. This feature depends on the edition of Microsoft PowerPoint in the user's computer. In Microsoft PowerPoint 2007 and Microsoft PowerPoint 2010, the feature will only show the list as an extension of the shortcut menu, but in the Microsoft PowerPoint 2013 will show all slide in thumbnail view.

The PowerPoint Controller has two non-java file that support the application, especially the *InputVoice* class, there are the grammar file, and the configuration file. The grammar file is code in a Java Speech Grammar File (JSGF) format, named as *input.gram* file and the configuration file is an xml document type. The Java Speech Grammar file consists of the grammar header, grammar name, and the grammar body. The grammar header format is:

#JSGF version char-encoding locale;

The "#JSGF" part is a declaration type file and it is essential while the rest of the line is optional. The grammar name declaration is mandatory and must be same with the grammar name defined in the configuration file. The format for the grammar name is: **grammar grammarName;**

The grammar body is the rule definition part where the rule can only declare once. If the rule declare more than once, the rule will be overwritten. The rule definition format is: **public <ruleName> = ruleExpansion;**

The ruleExpansion is a logical combination of tokens, and a reference to other rules. In other way, the ruleExpansion defines how the rule can be spoken. There is some optional symbol that can be used to vary the ruleExpansion. Table IV is described how the symbol works on a grammar file.

TABLE IV.
SYMBOL TO DEFINE RULE EXPANSION

Symbol	Example	Result of Grammar
- Alternatives	Hello Angel	Hello or Angel
() - Parentheses	(Hello Hi) (Angel Amy)	Hello Angel, or Hi Angel, or Hello Amy, or Hi Amy
[] - Optional grouping	[Hello] Angel	Hello Angel or Angel
* - Klenee star	(Hello Hi)* Angel	Hello Hi Hi Angel, or Hi Hi Hi Hi Angel, or Angel.
+ - Plus sign	(Hello Hi)+ Angel	Hello Hi Angel, or Hi Hi Angel, or Hi Angel. Exception: Angel.

The grammar for PowerPoint Controller named as *input* and the rule name is *greet*. This grammar construct from several words of input command and the alternative symbol. The purpose of this grammar is to provide the possible output word for recognition process in the *InputVoice* class. Figure 14 is an explanation code of *input.gram* file.

```
#JSGF V1.0;

grammar input;

public <greet> = (start here | open |
oven | often | close | lost | clause |
all slide | next | nest | nice | previous
| preface | refuse | mouse | most | enter
| better | up | of | down | done | left |
let | lap | right | write);
```

Figure 14. Code of JSGF Grammar

The configuration file is an xml format, named as *input.configuration.xml*. The configuration file give useful information for the speech recognition engine. The configuration file divided into several parts: the frequently tuned properties, the word recognizer configuration, the decoder configuration, the linguist configuration, the grammar configuration, the dictionary configuration, the miscellaneous configuration, the frontend configuration, the live front end configuration and the front end pipeline.

V. SYSTEM TESTING

A. Performance Evaluation

The PowerPoint Controller project will be evaluate for its performance in speech recognition. The performance evaluation will check if the application could recognize each of input command properly and execute the command to the system. However, the application performance have dependency on the user’s pronunciation and the environment. Table V explain the performance evaluation and its result.

TABLE V.
PERFORMANCE EVALUATION SCENARIO

Scenario	Expected Result	Evaluation
Open command	Start the PowerPoint slideshow from the beginning.	Work properly
Close command	Close the PowerPoint slideshow.	Work properly
Start here command	Start the PowerPoint slideshow from current slide.	Work properly
All slide command	Show menu to choose all slide in the slideshow.	Work properly
Next command	Change presentation to the next slide.	Work properly
Previous command	Change presentation to the previous slide.	Work properly
Right mouse click command	Show shortcut menu.	Work properly
Enter command	Perform enter button.	Work properly
Left command	Perform left button.	Work properly
Right command	Perform right button.	Work properly
Up command	Perform up button.	Work properly
Down command	Perform down button.	Work properly

B. Environment Evaluation

Regarding to the environment, the volume of input voice and the probability word. The environment specification will also give contribution for the result such as clear background voice, some other voice in distance 1 meter, a room with music voice, and the size of the room. The weight of this test will be 10 times for each command, so the whole weight of this test will be 120. The environment evaluation can be seen in Table VI.

TABLE VI.

ENVIRONMENT EVALUATION SCENARIO

Scenario Background	Test Weight	Test Result	Effective Result
Clear background	120	113	94.16%
Noise (30%)	120	100	83.33%
Noise (50%)	120	88	73.33%
Other voice in distance 1 m	120	110	91.66%
Music background (50% volume)	120	108	90.00%
Music background (70% volume)	120	102	85.00%
Music background (100% volume)	120	100	83.33%
Room size (6 m ²)	120	113	94.16%
Room size (9 m ²)	120	87	72.50%
Room size (12 m ²)	120	76	63.33%

The limitation of this research is it is only focus on implementing the speech recognition as an input for controlling Microsoft PowerPoint. The application will do the recognition using speech recognition library that implements the Hidden Markov Model in the system. The application will have some feature such as opening Microsoft PowerPoint application and adjustment feature. The adjustment feature is provided to give time for the microphone and application to adjust with the environment. The speaker dependency will not applied in this application, therefore this application will not cover feature for training the application by user’s voice.

The command that could be given in order to controlling the PowerPoint by the user are starting the presentation show, starting the presentation show from current slide, exiting the presentation show, perform option to choose slide, change slide to the next or previous slide, performing right mouse click, performing the arrow (up, down, left, right), and the enter button.

This research is not covering command for changing slide to the first and last slide, and the feature of opening specific Microsoft PowerPoint files from the application, microphone boost configuration and Bluetooth microphone setting.

VI. CONCLUSIONS

PowerPoint Controller can help people who need to control their PowerPoint slide during a presentation. The application is built under the Java framework as a desktop application. CMU Sphinx-4 was used for the ASR part. The application can run without dependency with the Bluetooth or internet connection. The application will control the PowerPoint slide based on the input voice given. In addition to its main function, PowerPoint Controller also has another feature such as the open Microsoft PowerPoint, adjustment menu, and the system tray menu.

The fundamental of PowerPoint Controller is the recognition and the controller. The recognition generates the input speech into a result text word, while the controller makes use of the result word to identify the controlling program to be executed. The adjustment menu has the same implementation for the recognition part but rather than sent the result word to the controller the adjustment menu will show the result text word in the adjustment menu window.

As a conclusion, PowerPoint Controller give benefit for the user who wants to made their presentation become more effective. The user can use this application from distant, using microphone or other input voice that compatible with their computer.

REFERENCES

- [1] Anusuya, M. A., and Katti, S. K. (2009). Speech recognition by machine: A review. *International Journal of Computer Science and Information Security, IJCSIS*, Vol. 6(3), pp. 181-205. Retrieved October 11, 2013, from arxiv.org: <http://arxiv.org/ftp/arxiv/papers/1001/1001.2267.pdf>
- [2] CMU Sphinx. Sphinx-4: A speech recognizer written entirely in the Java™ programming language. Retrieved October 11, 2013, from sourceforge.net: <http://cmusphinx.sourceforge.net/sphinx4/>
- [3] Gaskin, R. History of PowerPoint. Retrieved October 11, 2013, from robertgaskin.com: <http://www.robertgaskins.com/#powerpoint-history>
- [4] Jackson, M. (2005). Automatic Speech Recognition: Human Computer Interface for Kinyarwanda Language. Project Report Master of Science Degree in Computer Science of Makerere University. Retrieved October 11, 2013, from [www.fon.hum.uva.nl: http://www.fon.hum.uva.nl/IFA-publications/Others/Courses/MuhirweJackson_final_report-1.pdf](http://www.fon.hum.uva.nl/publications/Others/Courses/MuhirweJackson_final_report-1.pdf)
- [5] Lamere, P., et.al. 2003. Design of the CMU Sphinx-4 Decoder. Massachusetts: Mitsubishi Electric Research Laboratories, Inc.
- [6] Lawrence R. Rabiner, "A Tutorial on Hidden Markov Model and Selected Application in Speech Recognition", *Proceedings of the IEEE*, Vol. 77, No. 2, pp. 257 – 286, 1989.
- [7] Liao, C. F. Understanding the CMU Sphinx Speech Recognition System. Department of Computer Science, National Chengchi University. Retrieved October 11, 2013, from soa.csie.org: http://soa.csie.org/static-resources/homework/pr/PR_Final_Report.pdf
- [8] Mark Gales and Steve Young, "The Application of Hidden Markov Models in Speech Recognition", *Journal of Foundations and Trends in Signal Processing*, Vol. 1, No. 3, pp. 195–304, 2007
- [9] Microsoft. PowerPoint Keyboard Shortcut.[online].Retrieved October 11, 2013, from microsoft.com:<http://office.microsoft.com/en-001/powerpoint-help/keyboard-shortcuts-for-use-while-creating-a-presentation-in-powerpoint-2013-HA102749080.aspx>
- [10] Qystein Birkenes, Tomoko Matsui, Kunio Tanabe, Sabato Marco Siniscalchi, Tor Andre Myrvoll, and Magne Hallstein Johnsen, "Penalized Logistic Regression with HMM Log-Likelihood Regressors for Speech Recognition", *IEEE Transactions on Audio, Speech, and Language Processing* Vol. 18, No. 6, pp. 1440-1454, August 2010.
- [11] Rabiner, L. R. and Juang, B. H. 1986. An introduction to hidden markov models. *IEEE Magazine on Acoustics, Speech and Signal Processing*. 3(1) pp 4-16.
- [12] Xiong Xiao, Jinyu Li, Eng Siong Chng, Haizhou Li, Chin-Hui Lee, "A Study on Hidden Markov Models Generalization Capability for Speech Recognition", *Proceedings of ASRU-2009*, pp. 255-260, 2009.
- [13] Shun-Zheng Yu and Hisashi Kobayashi, "Practical Implementation of an Efficient Forward-Backward Algorithm for an Explicit-Duration Hidden Markov Model", *IEEE Transactions on Signal Processing*, Vol. 54, no. 5, pp. 1947-1955, May 2006.