Isolated Swahili Words Recognition using Sphinx4

Shadrack K. Kimutai, Edna Milgo, David Gichoya

Abstract— ***Speech recognition is one of the frontiers in Human Computer Interaction. A number of tools used to achieve speech recognition are currently available. One of such tools is Sphinx4 from Carnegie Mellon University (CMU). It has a recognition engine based on discrete Hidden Markov Model (dHMM) and a modular structure making it flexible to a diverse set of requirements. However, most efforts that have been undertaken using this tool are focused on established dialects such as English and French. Despite Swahili being a major spoken language in Africa, literature search indicates that little research has been undertaken in developing a speech recognition tool for this dialect. In this paper, we propose an approach to building a Swahili speech recognizer using Sphinx4 to demonstrate its adaptability to recognition of spoken Swahili words. To realize this, we examined the Swahili language structure and sound synthesis processes. Then, a 40 word Swahili acoustic model was built based on the observed language and sound structures using CMU Sphinxtrain and associate tools. The developed acoustic model was then tested using sphinx4.***

Keywords: Sphinx4, Swahili Language, Speech Recognition, Hidden Markov Model.

I. INTRODUCTION

Speech recognition has been an intense area of study. According to [1], researchers are seen approaching the field from various fronts of knowledge such as human sciences, statistics, artificial intelligence, linguistics, acoustic sciences, and information science amongst many other distinguished fronts. Speech can be termed as a subset of sound since it shares all the characteristics of sound. Indeed as observed by [2] human speech usually has a sampling rate that lie between 8Hz to 16 KHz. Speech can in this sense be redefined as a means through which information is relayed through air after its production in the human speech synthesizing organs.

Speech harbors many complex features that may not be easily realized unless under close scrutiny. These features include phones, phonemes, coagulation, and segmentation amongst others. Phones is the class of sounds numbering around fifty (50) which are used in all human languages. Phoneme on the other hand is the smallest unit of sound that has a distinct meaning. At this juncture it is appropriate to indicate that identification and documentation of Swahili phones is still going on with the total estimated number varying from 31 to 37 [3]. Articles [3] and [5] observed that Swahili dialect is made up of 32 phones with 5 being vowels. Swahili alphabet can be further grouped as follows: 23 single letters and 9 digraphs [3].

Manuscript received December 15, 2013.

Shadrack Kipchirchir Kimutai, IT Department, School of Information Science, Moi University, Eldoret, Kenya

Edna Milgo, IT Department, School of Information Science, Moi University, Eldoret, Kenya.

David Gichoya, IT Department, School of Information Science, Moi University, Eldoret, Kenya.

II. RELATED STUDIES

A number of researchers have made effort to develop an ASR's for the dialect. However, these efforts have encountered some challenges including lack of a standardized acoustic model for the dialect.

This has made research in this area costly. For instance, research in article [4] had to use crowdsourcing to construct their Acoustic model in the dialect. In their research, [4] justified the use of crowd sourcing was due to lack of acoustic model that suited their need. Another study is the Swahili-Text-To-Speech System by [6]. Still, their research did not deal with developing an ASR for the dialect but rather the research was limited to a study that resulted to the development of a system which could read out some Swahili text. Besides these, other co-related studies in the area are noticeable. One such study has undertaken a data driven 'part of speech' tagging which proves to be quite useful especially when working with bi-gram or tri-gram models [8]. Also related to this, [9] presented an effort that addresses a study on development of open source Spell-checking for Gikuyu dialect which is a Bantu dialect in Kenya It should be noted that both Bantu languages and Swahili shares a lot of morphological characteristics as identified by[5][6] and [9].

III. STATEMENT OF THE PROBLEM

Despite Swahili being a major spoken language in Africa, literature search reveals that little research has been undertaken in developing a speech recognition tool for this dialect. It is for this reason that we propose an effort of adapting Sphinx4 ASR for Swahili dialect.

Sphinx4 is capable of achieving speech recognition in any given language. However, the challenge of developing an acoustic model and language model of the language for any given dialect is left to any potential researcher who would like to develop an ASR. This is the case, especially when the dialect has not been adapted to any ASR.

IV. PROPOSED SOLUTION

In this paper we propose an approach to building a Swahili speech recognizer to demonstrate the adaptability of Sphinx4 to recognition of spoken Swahili words. To achieve this, we selected 40 words of which 31 were used as the training sample and the rest (9) used in testing sample. The 31 words were selected based on the phones they contain while the test sample was selected randomly ensuring that they cover the phones presented by the test sample. After the selection of these words, the multiple speech samples of each word was recorded. It is from these speech recordings that the training and development of the acoustic and language model was based on. Once developed, these models was plugged into sphinx4 recognition engine and tested.

V. SPHINX4

All ASR‟s in the Sphinx family utilize HMM for method of recognition. According to [12], the first member of the Sphinx family, Sphinx I, was built by Dr. Kai-fu Lee in 1987.It featured tri-phone support and word pair grammar. This ASR was soon after replaced by Sphinx II built by Dr. Xue Don Huang and was targeted to improving the rate of recognition an area which the first Generation of the recognizer had significant weakness [13]. The 3rd generation of the recognizer, Sphinx III was developed by Eric and Ravishankar and together with its derivative, Sphinx4, target users flexibility with the later adopting a modular architecture [12]. Walker et, al. in their article [14] states that Sphinx4‟s Frontend module may employ either Mel Frequency Cepstral Coefficient (MFCC) or Perceptual Linear Prediction coefficients (PLP) without altering the source code by configuring the module through the use of a xml. Sphinx4's pattern recognition approach has been clearly defined by [1] in which it composes of the steps as listed below.

A. Feature analysis and Optimization

According to [1] this initial stage concerns with formulation of appropriate spectral representation of a speech signal. In his research, [1] further states that the most widely used method of accomplishing this is through the use of a set/chain of filterbanks and linear predictive coding (LPC) or Vector quantization. However other researchers suggest methods can be used include Frequency Fourier Transform (FFT), Short Time Fourier transform (STFT) or Mel-Frequency Cepstral coefficients (MFCC) as discussed in article [7].

B. Unit matching system

According to [1], unit matching state involves searching the knowledge base of the recognizer for the most appropriate match. In Sphinx4 this involves building hidden Markov models for all the words based on the features obtained from step A above. Then the likelihood of all possible models is generated.

C. Selection of the best representation

Having matched all the models, the model whose likelihood is the highest (usually above a specific threshold) is deemed to be the best fit hence the word from which the model is generated from is deemed to have been pronounced.

VI. TOOLS AND SOFTWARE EMPLOYED IN THE STUDY

We employed the following tools and equipments:-

1) *Microphone*
2) *Sound card*
3) *Audacity*
4) *Ubuntu Operating Systems with Java Development Kit (JDK 1.7) installed*
5) *Eclipse Integrated Development Environment (IDE) (or any other IDE)*
6) *CMU Sphinx4*
7) *CMU SphinxTrain*
8) *CMU SphinxBase*
9) *CMU Clmtk tool kit*

VII. STEPS FOLLOWED IN THE RESEARCH

A. Selection of words

We selected 40 words from the Swahili dialect taking into consideration the representation of targeted phonetic sounds. Each word was then phonetically broken down into its constituent phones.

B. Recording of the selected words

We recorded the samples of speech of the selected words uttered by 6 different individuals 3 men and 3 women. We used a desktop microphone in an open environment and used Audacity to record the audio as seen in fig 1. Each word was recorded twice and tagged for unique identification of the speaker and the sample.

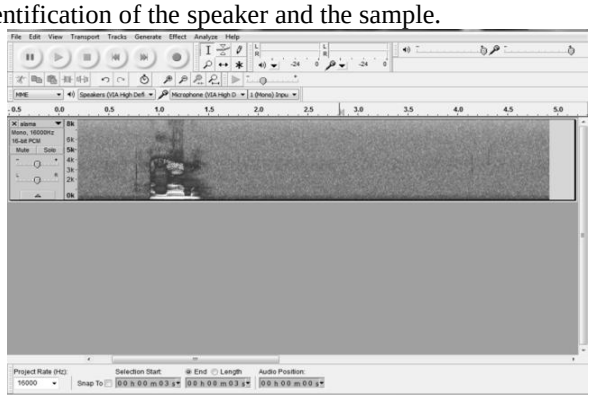

Fig 1: Screen Capture of the recording process of the word "ALAMA" using Audacity [15]

C. Development of a language model

We then used both CMU Clmtk and CMU Sphinx Base to develop a language model for the dialect. We began with the creation of a formatted source text in our case named "*kiswahili.text*" whose exerpt is shown below.

<*sil*> </*sil*>
<s> DAWA </s>
<s> SABALKHERI </s>
<s> WIMBI </s>
<s> ... </s>

This is then converted into a "*.wfreq*" file using the CMU Clmtk toolkit's wfreq tool by running the following commands.

text2wfreq < kiswahili.text > kiswahili.wfreq

Thereafter we generated a vocabulary file by converting the *.wfreq* file into a "*.vocab*" file using wfreq2vocab tool within the CMU Clmtk toolkit.

wfreq2vocab < kiswahili.wfreq > kiswahili.vocab

At this juncture, an *idngram* file was generated by the merger of both the "*.vocab*" and the "*.text*" files using the tool CMU Clmtk toolkit's text2idngram tool through the execution of the following code.

text2idngram -vocab kiswahili.vocab -idngram kiswahili.idngram < kiswahili.text

Finally, we generated an *.ngram* file from it that we obtained a trigram language model through execution of the code.

idngram2lm -vocab kiswahili.vocab -idngram kiswahili.idngram –arpa kiswahili.arpa

In order to use the developed trigram on sphinx it had to be converted into .DMP language

model format. This was accomplished by using Sphinx_lm_convert tool available in SphinxBase toolkit.

A. Development of an acoustic model

This step involved generation of acoustic models using sphinxbase and Sphinxtrain. We organized it such that Sphinxbase and Sphinxtrain where in the same folder, then we created the folder Kiswa3 to hold our acoustic model then from within it, executed the following commands.

~/Kiswa3 perl../sphinxtrain-1.0.7/
scripts_pl/setup_SphinxTrain.pl
-task Kiswa3

We thereafter obtained the paths to each wav file in wav folder and designated their respective paths and transcriptions in the following files for both the training set and the testing set.

1) *Kiswa3_test.fileids*
2) *Kiswa3_test.transcription*
3) *Kiswa3_train.fileids*
4) *Kiswa3_train.transcription*

After everything was set we generated the phone file and filler and then imported the .DMP language model file and the "*.arpa*" trigram language model file we had earlier generated. After this, the following commands were executed from the training folder

~/Kiswa3 perl ./scripts_pl/make_feats.pl -ctl etc/Kiswa3_train.fileids

~/Kiswa3 perl ./scripts_pl/make_feats.pl -ctl etc/Kiswa3_test.fileids

~/Kiswa3 perl ./scripts_pl/RunAll.pl

Final configuration and Deployment

Having obtained an acoustic model and a language model we now finalized its preparation by creating two files namely *config.xml* and *Kiswahili.java*. The *config.xml file* was used to pass configurations of the Frontend and also point to the acoustic model, language model, the filler and the dictionary to be used. *Kiswahili.java* on the other hand was to be the core file in the project. It imports input from the microphone, the sphinx4 class library and also loads the configurations from the *config.xml* to be used. After compilation, the resultant files (*Kiswahili.class*) and *config.xml* were added to a jar archive (*Kiswahili.jar*) before being used. Finally, we created an empty java project then importing sphinx4.jar, tags.jar, jsapi.jar and Kiswahili.jar and then using the Eclipse IDE project pane, navigating to the class and then executing the application.

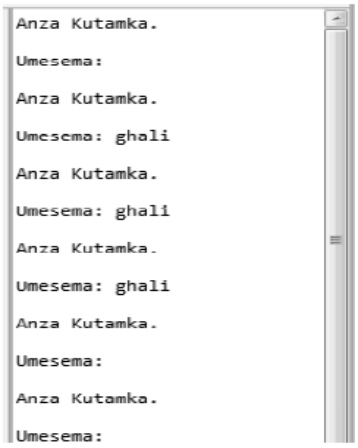

Fig 2: Screen Capture of the recognition process of the word Ghali. Note the two 'no classification' results.

VIII. ANALYSIS AND FINDINGS

After executing the sphinx4 with the generated acoustic models and language models, word utterances from each speaker were used to evaluate the recognizer. This evaluation involved uttering training, testing samples used in the generation of the acoustic model. Results of each utterance were recorded and grouped into three categories; correct classification, wrong classification and no classification. Table I summarizes the findings.

From the **Table 1**, the aggregate recognition rate was deduced to be 53% with 107 correct classifications out of a total of 200 pronunciations from a microphone. We noted, however that no classification rates were higher than anticipated.

WORD	CORRECT CLASSIFICATION CC	WRONG CLASSIFICATION WC	NO CLASSIFICATION NC	WORD RECOGNITION RATE $WRR = \frac{[CC - (WC + NC)]}{5} \times 100$
ALAMA	2	0	3	40
BAO	3	2	0	60
CHAMA	3	1	1	60
CHUO	0	2	3	0
DAWA	5	0	0	100
DEBE	2	2	1	40
DHAMBI	3	0	2	60
FAIDHA	2	0	3	40
GHALI	3	0	2	60
GOLI	3	2	0	60
HELA	3	0	2	60
JINO	2	1	2	40
KANGA	2	2	1	40
LIMA	1	4	0	20
MAMBA	3	2	0	60
MASHARTI	3	1	1	60
MOI	3	1	1	40
MZIZI	2	3	0	40
NENO	3	1	1	60
NGOMBE	3	0	2	60
NOA	2	2	1	40
NYOTA	3	1	1	60
PANGO	2	3	0	40
PAPA	3	1	1	60
RAHISI	4	1	0	80
RUNINGA	3	0	2	60
SABALKHERI	4	1	0	80
SARUFI	3	2	0	60
SARUJI	2	3	0	40
SHEMEJI	3	1	1	60
THELUJI	4	0	1	80

Table 1: A summary of Selected Words Recognition

IX. CONCLUSION

From the analysis it is clear that the recognition rates we achieved where lower compared to findings by other researchers such as Satori et al, [12] who had employed the same tool. However, we were able to deduce that if we repeated every 'no classification' as shown in **fig 2**, our aggregate recognition rate might have fared well above our findings. Furthermore we

were pleased by the recognition rates of unmodelled words. In our future work, we will employ this tool to explore expanded acoustic model with more samples to train on and also explore Swahili continuous speech recognition using the tool. Besides this, we are on how to integrate Sphinx4 with neural network ASR's.

ACKNOWLEDGMENT

The Authors would like to thank the Moi University School of Information Sciences staff and students who contributed to this research. Furthermore, The Authors would like to thank members of CMU Sphinx Forum for guidance regarding the tool.

REFERENCES

1. Juang B H, and L R Rabiner "Fundamentals of speech recognition" Eaglewood Cliffs, New Jeysey: PTR-Prentice Hill,1993.
2. Rusell Stuart and Peter Norvig, "Artificial intelligence: A modern approach 3rd Edition" Upper Saddle River, New Jeysey; Pearson Education, inc, 2010.
3. Choge, Susan. "Understanding Kiswahili Vowels." The Journal of Pan African Studies, 2009: 2.8, 62-77.
4. Hadrien, Gelas, Teferraabate Solomon, Besacier Laurent, and Pellegrino François. Quality assessment of crowdsourcing transcriptions for African languages. Lyon: Universit´e de Lyon, 2010.
5. Mwasimba. "Chapter 1 - Swahili Spelling and Pronunciation." Mwasimba Online. April 4, 2009.Available [online]: http://mwanasimba.online.fr/E_Chap01.htm accessed 13th,September 2011.
6. Ngugi, K,W Okelo and P, Wagacha, "Swahili text to speech system", African Journal of Science and Technology,49.1,88-89.
7. Huang, Xuedong, Alex Acero, and Hon Hsiao-wuen. Spoken Language Processing. Upper Saddle River, New Jersey: Prentice-Hall Inc, 2001).
8. Guy, De Pauw, De Schryver Gilles-Maurice, and W Wagacha Peter. "Data-Driven Part-of-Speech Tagging of Kiswahili." Text, speech and dialogue, ninth international conference, Proceedings. Berlin, Germany: Springer, 2006.
9. Chege, K., Wagacha, P. W., De Pauw Guy, M. L., & Wanjiku, N. (2010). "Developing an Open source spell checker for Gĩkũyũ". Second Workshop on African Language Technology . II. Valletta, Malta: European Language Resources Association (ELRA).
10. Carnegie Mellon University. Sphinx4 [Online]. Available: http://cmusphinx.sourceforge.net.
11. Liao, Chun-Feng. "Understanding the CMU Sphinx Speech Recognition System." Taipei: National Chengchi University, 2003: 17-22
12. H. Satori, M Harti and N Chenfour. "Introduction to Arabic Speech Recognition Using CMUSphinx System". Information and Computer Science. pp.173, 115, 2010.
13. Huang, Xuedong,Alleva Fileno,Hwang Mei-Yuh and Rosenfeld Ronald "An overview of the SPHINX-II speech recognition system" Computer,Speech and Language,1992 7.1,137-148
14. Willie, Walker, et al. Sphinx-4: A Flexible Open Source Framework for Speech Recognition. Sun Microsystems inc., 2004.
15. Audacity [Online] Available: http://audacity.sourceforge.net

AUTHORS PROFILE

Shadrack K Kimutai Shadrack is an M.phil Information Technology student at School of Information Sciences, Moi University. He also holds a B.sc Computer Science from Kabarak University. He has published conference paper and currently researching integration of sphinx4 and neural networks. His research interests are artificial intelligence, computer graphics and areas of information systems.

Edna Milgo .Edna is a Lecturer of Information Technology at Moi University. She holds an MSc. Computer Science from Columbus State University(USA) and Bsc. Computer Science from Kenyatta University(Kenya). Her research interests are Artificial Intelligence, Machine learning and statistical tools for Information Security. Her publications are in the areas of security and artificial intelligence. Edna received a Honor and membership to Phi Kappa Phi Society(2009), Who's Who Among American Students(2010), and Second Position-Masters category: -Mid-Southeast Chapter of ACM(2008).

David Gichoya David is a Senior Lecturer in the Department of Information Technology, Moi University (MU). He holds a PhD in information and Computer Science from Loughborough University (UK). He Studied Msc in Computer Science and Application at Shanghai University (China). Amongst his numerous publications, He has authored a book entitled "Towards successful implementation of ICT projects in developing countries"