International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 8, ISSN No. 2455-2143, Pages 312-315
Published Online December 2019 in IJEAST (http://www.ijeast.com)

WAKE-UP-WORD SYSTEM USING SPHINX-4

Gamal Bohouta
Department of Computer Engineering and Sciences
Florida Institute of Technology, Melbourne, FL, USA

Veton Këpuska
Department of Computer Engineering and Sciences
Florida Institute of Technology, Melbourne, FL, USA

***Abstract*— Many techniques have focused on improving the accuracy of speech recognition in General ASR systems, which are better known as Large-Vocabulary Continuous Speech Recognition systems or General ASR systems (General ASR). Some other approaches have focused on Wake-Up-Word ASR systems (WUW ASR), which are similar to Key-Word spotting. One important aspect of WUW ASR systems is the ability to discriminate the specific word or phrase used only in an alerting context and not in other, referential contexts. However, the aim of this paper is to evaluate the WUW approach by implementing a new WUW ASR system that can be used to recognize between the word in alerting context or referential contexts. Furthermore, the new ASR system will be able to reduce the number of false alarms in the devices and applications that use the speech commands to activate the devices and applications.**

Keywords— **Speech Recognition, Wake-Up-Word, Support Vector Machine, Sphinx-4**

I. Introduction

The WUW ASR, which is similar to Key-Word spotting, is one important technology used for discrimination the word/phrase used only in alerting context and not in the referential contexts [1]. Most companies that produce ASR systems have focused on improving the speech recognition accuracy in General ASR systems without improving the speech recognition accuracy in WUW ASR systems[6]. Recently, WUW system has come to the forefront of speech recognition with the advent of voice-assist technologies such as Microsoft Cortana, Amazon Alexa, Apple Siri, and Google Assistant[7]. All of these companies have started focus on the WUW technology to improve the wake words that activate their devices for interaction with the users. The applications or devices that use WUW ASR can discriminate between the word in alerting context or referential contexts when a user speaks a word like “Computer" in alerting context such as "Computer, show me the chart? " and in referential context, “Every computer should have a speaker”. On the other hand, the applications or devices that use General ASR will not be able to discriminate between the two cases. It is very hard to determine in real-time if the user is speaking to the computer or about the computer. In other words, the WUW ASR should be able to discriminate if the user is speaking to the recognizer or not. However, this paper aims to use the WUW approach to implement the new WUW ASR system that can be used to recognize between the word in alerting context or referential contexts. Furthermore, it will be used in applications that will need high accuracy WUW ASR system.

Moreover, this paper intrudes the steps that have been followed to evaluate and test the new WUW system to ensure that the WUW approach is the best way to detect WUW, which is in of vocabulary words (IOV) with high accuracy and reject the non-WUW, which is out of vocabulary words (OOV). In order to design the new WUW system and understand all WUW ASR system components, the following steps were taken (1) choosing ASR system, (2) applying the WUW approach, (3) testing the new WUW system. In this study, Sphinx 4 was selected to test the WUW approach Sphinx 4 was selected to test the WUW approach that has defined and investigated by Veton Kepuska and his team.

The structure of the WUW approach includes three major components, which are (1) Front End: responsible for features extraction and VAD classification of each frame. (2) Back End: performing word segmentation and classification of those segments for each feature stream, and (3) INV/OOV Classification using individual HMM segmental scores with SVM[1].

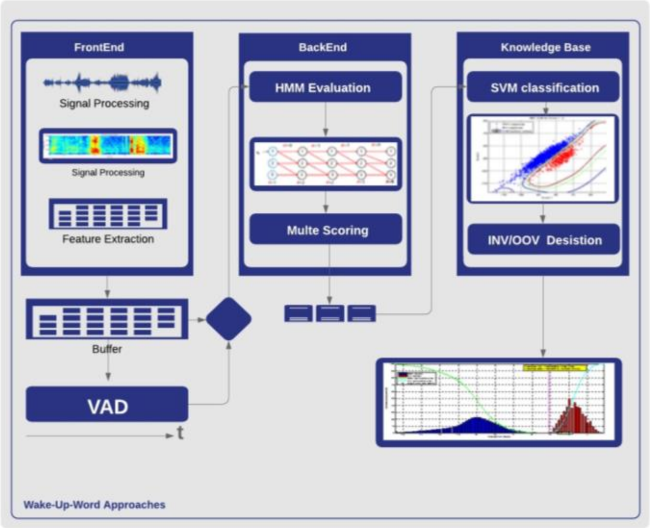


Fig. 1. The Structure of the WUW Approach.

II. WAKE-UP-WORD PERFORMANCE

The goal of this paper is to test the WUW approach to ensure that it is the best way to detect WUW, which is in of vocabulary words IOV with high accuracy and reject the non-WUW, which is out of vocabulary words OOV. Moreover, to study the WUW approach can discriminate between the word in alerting context or referential contexts, such as in the use of the word "Computer" in alerting context is " Can you solve this, Computer? " and in referential context it is " A Computer is normally used in presentation". The following steps were taken in order to test the approach: choosing ASR system, applying Wake-up-word approach, Testing the system.

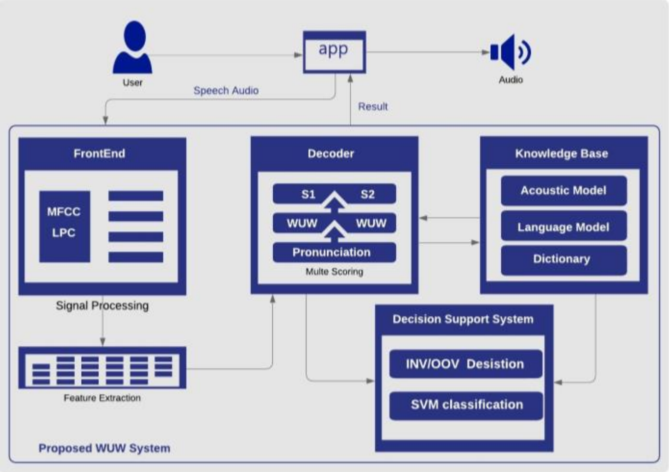


Fig. 2. The structure of the New WUW System

A. Choosing ASR system

Choosing a suitable platform for testing is an important step to ensure the proposed WUW system can work with high accuracy. Today, there are several companies using ASR systems in their products, such as Amazon, Microsoft, Google, Sphinx-4, HTK, Kaldi and Dragon [2]. In our study, Sphinx-4 was selected for testing the approach based on its supporting, open-source system, programming language, and structure of components. There are four main reasons for choosing the Sphinx-4. The first reason is Sphinx-4 developed at Carnegie Mellon University (CMU) and currently has an extensive vocabulary speaker independent speech recognition, and its source code is available for download and use [5]. The second reason is "Sphinx-4 is an open source speech recognition system that incorporates state-of-the art methodologies and also addresses the needs of emerging research areas" [4]. The third reason is its structure has three main components, which is the same as our structure. Its structure includes the Frontend, the Decoder, and the Linguist. Moreover, its structure was designed with a high level of suppleness and modularity [3]. The fourth reason is the Sphinx-4 was written in the Java programming language. Therefore, there are additional packages such as Pocketsphinx, Sphinxbase, and Sphinx-train that can be used to train and test the acoustic model.

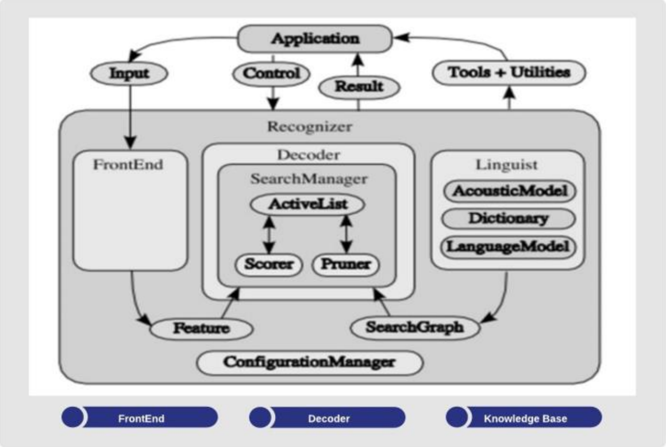


Fig. 3. The Structure of the Sphinx-4

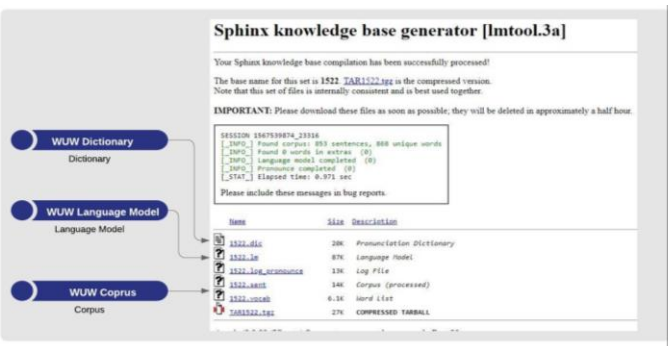
B. Applying Wake-up-word approach

There are some steps that have been followed to apply the WUW approach with Spinx-4. For example, (1) The Frontend of WUW speech recognition should be able to accept various sets of features such as MFCC, LPC, and ENH-MFCC; (2) The features are needed to be decoded and reversed with corresponding HMMs in the back-end stage of the WUW speech recognizer; (3) Applying different features to the decoder to get different scores from the acoustic model. (4) using the Support Vector Machine (SVM) system to help the system for making the final decision if the result is IOV or OOV based of the Scores of Acoustic Models.

Using Sphinx-4 to create two kinds of features, the first feature is stander MFCC or LPC features that has been generated by using the stander of Sphinx-4 Frontend. The stander of Sphinx-4 Front End can generate two types of features MFCC and LPC. The second feature that has been added to the stander of Sphinx-4 Frontend is reversed the stander of Sphinx-4 Front End. This feature uses to get the second score that will be compared with the first score to make the final decision. Moreover, The WUW acoustic model was created using the Pocketsphinx, Sphinxbase, and Sphinxtrain with other languages Perl, Python.

Also, The WUW acoustic model was trained by using the WUW corpus that collected by speech recognition group at Florida Institute of Technology or from the Speech Commands Dataset. In this part, only the alerting contexts was used to train the WUW acoustic model. Also, the stander and reverse features have been applied to the WUW ASR system to get different scores from the acoustic models, WUW Score1 from the WUW acoustic model with stander features (WUW1: -5.19E+07) and WUW Score2 from the WUW acoustic model with reverse features (WUW1: -5.19E+07). Also, by using the

Support Vector Machine (SVM) library which is JAVA LIBSVM [11] with Sphinx-4, the system can make the final decision that if the result is IOV or OOV and detect a WUW word or phrase while rejecting all other words or sounds based of the score of WUW acoustic models (Score 1) and (Score 2).



Sphinx knowledge base generator [lmtool.3a]

Your Sphinx knowledge base compilation has been successfully processed!

The base name for this set is **1522**. TAR1522.tgz is the compressed version.
Note that this set of files is internally consistent and is best used together.

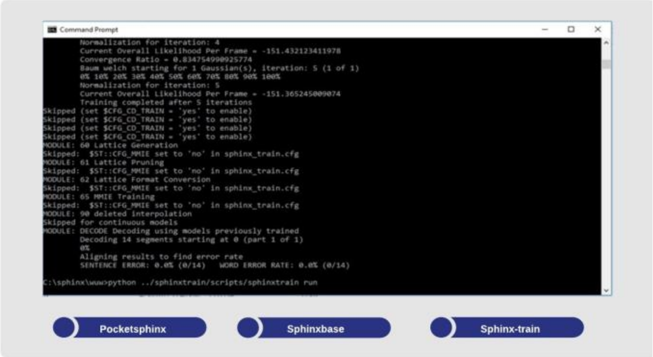
IMPORTANT: Please download these files as soon as possible; they will be deleted in approximately a half hour.

```
SESSION 1567539874_23316
[_INFO_] Found corpus: 853 sentences, 888 unique words
[_INFO_] Found 0 words in extras  (0)
[_INFO_] Language model completed  (0)
[_INFO_] Pronounce completed  (0)
[_STAT_] Elapsed time: 0.971 sec
```

Please include these messages in bug reports.

Name	Size	Description
1522.dic	28K	Pronunciation Dictionary
1522.lm	87K	Language Model
1522.log_pronounce	13K	Log File
1522.sent	14K	Corpus (processed)
1522.vocab	6.1K	Word List
TAR1522.tgz	27K	COMPRESSED TARBALL

Fig. 4. Sphinx Knowledge Base LMtool



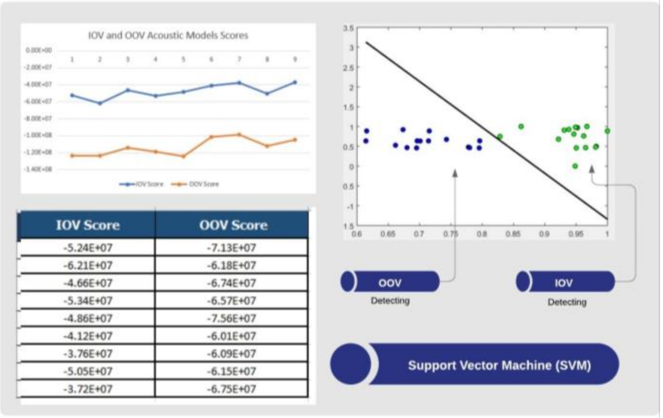
```
        Normalization for iteration: 4
        Current Overall Likelihood Per Frame = -151.432123411978
        Convergence Ratio = 0.834754990925774
        Baum welch starting for 1 Gaussian(s), iteration: 5 (1 of 1)
        0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
        Normalization for iteration: 5
        Current Overall Likelihood Per Frame = -151.365245009074
        Training completed after 5 iterations
Skipped (set $CFG_CD_TRAIN = 'yes' to enable)
Skipped (set $CFG_CD_TRAIN = 'yes' to enable)
Skipped (set $CFG_CD_TRAIN = 'yes' to enable)
Skipped (set $CFG_CD_TRAIN = 'yes' to enable)
MODULE: 60 Lattice Generation
Skipped:  $ST::CFG_MMIE set to 'no' in sphinx_train.cfg
MODULE: 61 Lattice Pruning
Skipped:  $ST::CFG_MMIE set to 'no' in sphinx_train.cfg
MODULE: 62 Lattice Format Conversion
Skipped:  $ST::CFG_MMIE set to 'no' in sphinx_train.cfg
MODULE: 65 MMIE Training
Skipped:  $ST::CFG_MMIE set to 'no' in sphinx_train.cfg
MODULE: 90 deleted interpolation
Skipped for continuous models
MODULE: DECODE Decoding using models previously trained
        Decoding 14 segments starting at 0 (part 1 of 1)
        0%
        Aligning results to find error rate
        SENTENCE ERROR: 0.0% (0/14)   WORD ERROR RATE: 0.0% (0/14)

C:\sphinx\wuw>python ../sphinxtrain/scripts/sphinxtrain run
```

Fig. 5. Training and Testing the Acoustic Models

C. Testing the new WUW ASR system

After testing the new approach with Sphinx-4, for the corpus and creating two features, the new ASR system was able to make the final decision based on the results of the decoder and the scoring of the WUW acoustic models. In the stage, The Support Vector Machine (SVM) library with the WUW system was used to detect a single word or phrase while rejecting all other words or sounds by using the score of WUW acoustic models (Score1) and the score of reverse WUW acoustic models (Score2). For evaluating the detection of the WUW, we tested some corpus with the support vector machine. For evaluating the system, it was tested with some corpus. The results of the experiment showed that the WUW ASR system can make the final decision if the result is IOV word with high accuracy (Confidence Word (100%)) OOV word with Confidence Word (100%). The following figure illustrates the experiment results:



IOV Score	OOV Score
-5.24E+07	-7.13E+07
-6.21E+07	-6.18E+07
-4.66E+07	-6.74E+07
-5.34E+07	-6.57E+07
-4.86E+07	-7.56E+07
-4.12E+07	-6.01E+07
-3.76E+07	-6.09E+07
-5.05E+07	-6.15E+07
-3.72E+07	-6.75E+07

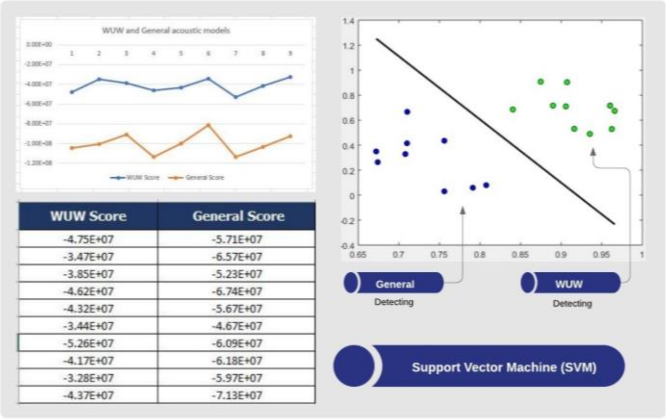
Fig. 6. IOV and OOV Acoustic Models Scores

III. EXPERIMENT AND RESULT

In order to test the performance of the new system in different acoustic environments, the WUW ASR system has been training and testing with different acoustic environments, such as different background noise levels, different speaker distances to the microphone, and various speakers. The WUW ASR system was tested with four acoustic environments: testing the WUW system with different noise types, such as door slam noise, white noise, babble noise, pink noise, blue noise, red noise, and violet noise; testing the WUW system with noise levels that vary in 5dB steps and ranges from 5dB to 50dB; testing the WUW system when the speaker is positioned far from or close to a microphone in a recording. Finally, testing the WUW system by the speakers of several major dialects of English.

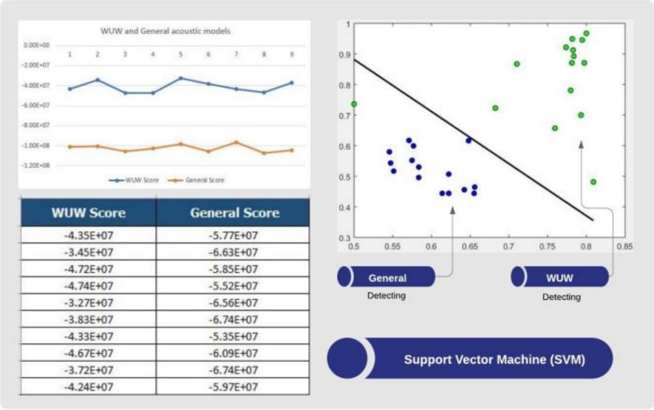
Many types of speech data were selected from various sources to train and evaluate the proposed ASR system. Speech data collected from different sources were used to produce and train the acoustic models. Other speech data were used to test the performance of the WUW ASR system with characteristics of acoustic environments, such as noise levels, speaker accents, and microphone variability. The speech data used with the proposed ASR system are the TIMIT corpus, WUW corpus, Speech Commands Dataset, FMTIMIT corpus, the Noisy TIMIT Speech corpus, and Speech Commands Dataset Background Noise corpus[8][9][10].

Based on experimental results, the ASR performs well, with high quality, with high performance in all acoustic environments and in all stages of the proposed system. Also, the system can detect WUW or non-WUW with high accuracy (Confidence Word 100%).



WUW Score	General Score
-4.75E+07	-5.71E+07
-3.47E+07	-6.57E+07
-3.85E+07	-5.23E+07
-4.62E+07	-6.74E+07
-4.32E+07	-5.67E+07
-3.44E+07	-4.67E+07
-5.26E+07	-6.09E+07
-4.17E+07	-6.18E+07
-3.28E+07	-5.97E+07
-4.37E+07	-7.13E+07

Fig. 7. Using Pink Noise with the WUW system



WUW Score	General Score
-4.35E+07	-5.77E+07
-3.45E+07	-6.63E+07
-4.72E+07	-5.85E+07
-4.74E+07	-5.52E+07
-3.27E+07	-6.56E+07
-3.83E+07	-6.74E+07
-4.33E+07	-5.35E+07
-4.67E+07	-6.09E+07
-3.72E+07	-6.74E+07
-4.24E+07	-5.97E+07

Fig. 8. Using FMTIMIT Corpus with the WUW system

IV. CONCLUSION

The specific objective of this study is to address the stages for building a WUW ASR system that can be used with our proposed ASR system and to report the experiments that have been followed to test the performance of WUW ASR system. The WUW ASR system has been training and testing with many types of speech corpora contain different acoustic environments. Based on experimental results, It is obvious that the WUW ASR performance is work with high quilt for all acoustic environments and all stages of proposed WUWASR system work with high performance. Also, the system can detect the WUW with high accuracy (Confidence Word (100%)) or General word with Confidence Word (100%).

V. REFERENCE

[1] V. Këpuska, T. Klein (2009), "A novel Wake-Up-Word speech recognition system, Wake-Up-Word recognition task, technology and evaluation", Nonlinear Analysis *Elsevier*.

[2] V.Këpuska, G. Bohouta (2017), "Comparing Speech Recognition Systems (Microsoft API, Google API And CMU Sphinx)", Int. J. of Engineering Research and Application, vol. 7, pp. 20-24, March, 2017

[3] P. Lamere, P. Kwok, E. Gouvêa, B. Raj, R. Singh, W. Walker, M. Warmuth, Peter Wolf (2002), "The Cmu Sphinx-4 Speech Recognition System", Sun Microsystems Laboratories,Carnegie Mellon University,Mitsubishi Electric Research Labs and University of California, USA.

[4] E. Babu, S. Jeelan and P. Prakash (2013), "Static dictionary for Telugu speech recognition system", Int. J. of Conceptions on Computing and Information Technology, vol. 1, pp. 29-32, November, 2013

[5] P. Lamere, P. Kwok, E. Gouvêa, B. Raj, R. Singh, W. Walker, M. Warmuth, Peter Wolf (2004), " Sphinx-4: A Flexible Open Source Framework for Speech Recognition", Sun Microsoft, USA, November 2004.

[6] V. Këpuska, G. Bohouta (2017), "Improving Wake-Up-Word and General Speech Recognition Systems", : 2017 IEEE Cyber Science and Technology Congress.

[7] V. Këpuska, G. Bohouta (2018), "Next-Generation of Virtual Personal Assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home), IEEE CCWC 2018.

[8] A. Abdulaziz, V. Kepuska (2018), "Noisy TIMIT Speech", Linguistic Data Consortium 2017

[9] Google , "Launching the Speech Commands Dataset", Google AI 2017

[10] TIMIT Acoustic-Phonetic Continuous Speech Corpus. Linguistic Data Consortium:
https://catalog.ldc.upenn.edu/LDC93S

[11] C. Chang and C. Lin, LIBSVM (2011), " a library for support vector machines". ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011. Software available at
http://www.csie.ntu.edu.tw/~cjlin/libsvm