RESEARCH ARTICLE OPEN ACCESS

Comparing Speech Recognition Systems (Microsoft API, Google API And CMU Sphinx)

Veton Këpuska[1], Gamal Bohouta[2]
[1,2](Electrical & Computer Engineering Department, Florida Institute of Technology, Melbourne, FL, USA

ABSTRACT
The idea of this paper is to design a tool that will be used to test and compare commercial speech recognition systems, such as Microsoft Speech API and Google Speech API, with open-source speech recognition systems such as Sphinx-4. The best way to compare automatic speech recognition systems in different environments is by using some audio recordings that were selected from different sources and calculating the word error rate (WER). Although the WER of the three aforementioned systems were acceptable, it was observed that the Google API is superior.

Keywords: Speech Recognition, Testing Speech Recognition Systems, Microsoft Speech API, Google Speech API, CMU Sphinx-4 Speech Recognition.

I. INTRODUCTION

Automatic Speech Recognition (ASR) is commonly employed in everyday applications. "One of the goals of speech recognition is to allow natural communication between humans and computers via speech, where natural implies similarity to the ways humans interact with each other" [8]. ASR has provided many systems that have been used to increase the interaction experience between users and computers. According to Dale Isaacs, "Today automatic speech recognition (ASR) systems and text-to-speech (TTS) systems are quite well established. These systems, using the latest technologies, are operating at accuracies in excess of 90%" [6]. Due to the increasing number of ASR systems, such as Microsoft, Google, Sphinx, WUW, HTK and Dragon, it becomes very difficult to know which of them we need. However, this paper shows the results of testing Microsoft API, Google API, and Sphinx4 by using a tool that has been designed and implemented using Java language with some audio recordings that were selected from a large number of sources. Also, in comparing those systems a number of various components were utilized and evaluated such as the acoustic model, the language model, and the dictionary.

There are a number of commercial and open-source systems such as AT&T Watson, Microsoft API Speech, Google Speech API, Amazon Alexa API, Nuance Recognizer, WUW, HTK and Dragon [2]. Three systems were selected for our evaluation in different environments: Microsoft API, Google API, and Sphinx-4 automatic speech recognition systems. Two of the biggest companies building voice-powered applications are Google and Microsoft [4]. The Microsoft API and Google API are the commercial speech recognition systems whose code is inaccessible, and Sphinx-4 is one of the ASR systems whose code is freely available for download [3].

II. THE CMU SPHINX

The Sphinx system has been developed at Carnegie Mellon University (CMU). Currently," CMU Sphinx has a large vocabulary, speaker independent speech recognition codebase, and its code is available for download and use" [13]. The Sphinx has several versions and packages for different tasks and applications such as Sphinx-2, Sphinx-3 and Sphinx-4. Also, there are additional packages such as Pocketsphinx, Sphinxbase, Sphinxtrain. In this paper, the Sphinx-4 will be evaluated. The Sphinx-4 has been written by Java programming language. Moreover," its structure has been designed with a high degree of flexibility and modularity" [13]. According to Juraj Kačur, "The latest Sphinx-4 is written in JAVA, and Main theoretical improvements are: support for finite grammar called Java Speech API grammar, it doesn't impose the restriction using the same structure for all models" [13] [5]. There are three main components in the Sphinx-4 structure, which includes the Frontend, the Decoder and the Linguist. According to Willie Walker and other who have worked in Sphinx-4, "we created a number of differing implementations for each module in the framework. For example, the Frontend implementations support MFCC, PLP, and LPC feature extraction; the Linguist implementations support a variety of language models, including CFGs, FSTs, and N-Grams; and the Decoder supports a variety of Search Manager implementations" [1]. Therefore, Sphinx-4 has the most recent version of an HMM-based speech and a

strong acoustic model by using HHM model with training large vocabulary [2].

III. THE GOOGLE API

Google has improved its speech recognition by using a new technology in many applications with the Google App such as Goog411, Voice Search on mobile, Voice Actions, Voice Input (spoken input to keypad), Android Developer APIs, Voice Search on desktop, YouTube transcription and Translate, Navigate, TTS.

After Google, has used the new technology that is the deep learning neural networks, Google achieved an 8 percent error rate in 2015 that is reduction of more than 23 percent from year 2013. According to Pichai, senior vice president of Android, Chrome, and Apps at Google, "We have the best investments in machine learning over the past many years. Indeed, Google has acquired several deep learning companies over the years, including DeepMind, DNNresearch, and Jetpac"[11].

IV. THE MICROSOFT API

Microsoft has developed the Speech API since 1993, the company hired Xuedong (XD) Huang, Fil Alleva, and Mei-Yuh Hwang "three of the four people responsible for the Carnegie Mellon University Sphinx-II speech recognition system, which achieved fame in the speech world in 1992 due to its unprecedented accuracy. the first Speech API is (SAPI) 1.0 team in 1994" [12].

Microsoft has continued to develop the powerful speech API and has released a series of increasingly powerful speech platforms. The Microsoft team has released the Speech API (SAPI) 5.3 with Windows Vista which was very powerful and useful. On the developer front, "Windows Vista includes a new WinFX® namespace, System.Speech. This allows developers to easily speech-enable Windows Forms applications and apps based on the Windows Presentation Framework"[12].

Microsoft has focused on increasing emphasis on speech recognition systems and improved the Speech API (SAPI) by using a context-dependent deep neural network hidden Markov model (CD-DNN-HMM). According to the researchers who have worked with Microsoft to improve the Speech API and the CD-DNN-HMM models, they determined that the large-vocabulary speech recognition that achieves substantially better results than a Context-Dependent Gaussian Mixture Model Hidden Markov mode12]. Just recently Microsoft announced "Historic Achievement: Microsoft researchers reach human parity in conversational speech recognition" [15].

V. EXPERIMENTS

The best way to test the quality of various ASR systems is to calculate the word error rate (WER). According to the WER, we can also test the different models in the ASR systems, such as the acoustic model, the language model, and the dictionary size. However, in this paper we have developed a tool that we have used to test these models in Microsoft API, Google API, and Sphinx-4. Also, we have calculated the WER by using this tool to recognize a list of sentences, which we collected in the form of audio files and text translation. In this paper, we follow these steps to design the tool and test Microsoft API, Google API, and Sphinx-4.

VI. TESTING DATA

The audio files were selected from various sources to evaluate the Microsoft API, Google API, and Sphinx-4. According to CMUSphin, Sphinx-4's decoder supports only one of the two specific audio formats (16000 Hz / 8000 Hz) [13]. Also, Google does not recognize the WAV format generally used with Sphinx-4. Part of the process of recognizing WAV files with Google involves converting the WAV files to the FLAC format. Microsoft can recognize any WAV files format. However, we solved this problem by making our tool recognize all audio files in the same format (16000 Hz / 8000 Hz).

Some of the audio files have been selected from the TIMIT corpus." The TIMIT corpus of read speech is designed to provide speech data for acoustic-phonetic studies and for the development and evaluation of automatic speech recognition systems. TIMIT contains broadband recordings of 630 speakers of eight major dialects of American English, each reading ten phonetically rich sentences" [14]. "The TIMIT corpus includes time-aligned orthographic, phonetic and word transcriptions as well as a 16-bit, 16kHz speech waveform file for each utterance. Corpus design was a joint effort among the Massachusetts Institute of Technology (MIT), SRI International (SRI) and Texas Instruments, Inc. (TI)" [9].

Also, we have selected other audio files from ITU (International Telecommunication Union) which is the United Nations Specialized Agency in the field of telecommunications [10]. Example of some of the audio files are presented in the table1 below:

File	The original sentences
SX293	Please take this dirty table cloth to the cleaners for me.
SX223	Put the butcher-block table in the garage.
SI1894	My father ran him off here six years ago.
SI1400	Now that this is at odds with our meaning may be shown as follows.
SX188	Who authorized the unlimited expense account?
SI1628	This is my hen ledger, he informed him in an absorbed way.
SI2000	We can get it if we dig, he said patiently.
SX216	The small boy put the worm on the hook.
SX396	The fish began to leap frantically on the surface of the small lake.
SI1580	He always seemed to have money in his pocket.
SX209	Michael colored the bedroom wall with crayons.
SI1584	Rector was often curious; often tempted to ask questions but he never did.
SX371	Right now may not be the best time for business mergers.
SI1373	Each form represented by the dictionary is looked up in the text form list
SX233	Highway and freeway mean the same thing.
AENGM8	The coffee stand is too high for the couch.
AENGF8	His hip struck the knee of the next player.
AENGF7	Grape juice and water mix well.
AENGM2	Jazz and swing fans like fast music

Table 1. The Audio Files

VII. SYSTEM DESCRIPTION

This system has been designed by using the Java language, which is the same language that has been used in Sphinx-4, as well as the C# that was used to test the Microsoft API and Google API. Also, we have used several libraries such as Text to Speech API, Graph API and Math API for different tasks. Moreover, this tool was connected with the classes of Sphinx4, Microsoft API and Google API to work together to recognize the audio files. Then we compared the recognition results with the original recording texts.

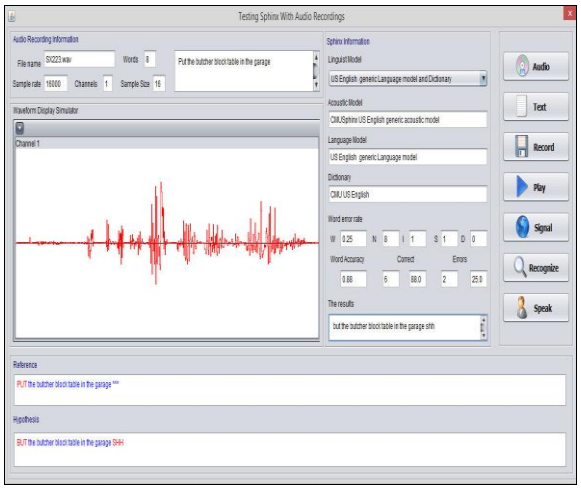

Figure 1. The System Interface.

VIII. EXPERIMENTAL RESULTS

The audio recordings with the original sentences were used to test the Sphinx-4, Microsoft API, and Google API. By using our tool, we have tested all files and calculated the word error rate (WER) and accuracy. We calculated the word error rate (WER) and accuracy according to these equations.

WER = (I + D + S) / N
WER = (0 + 0 + 1) / 9 = 0.11

where I words were inserted, D words were deleted, and S words were substituted.
The original text (Reference):
the small boy **PUT** the worm on the hook
The recognition text (Hypothesis):
the small boy **THAT** the worm on the hook

Accuracy = (N - D - S) / N
WA = (9 + 0 + 1) / 9 = 0.88

The original text (Reference):
the coffee **STANDARD** is too high for the couch
The recognition text (Hypothesis):
the coffee **STAND** is too high for the couch

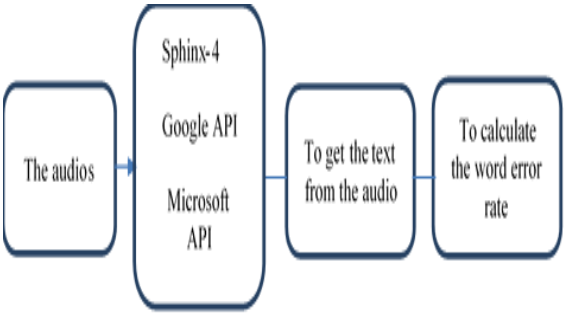

Figure 2. The Structure of The System.

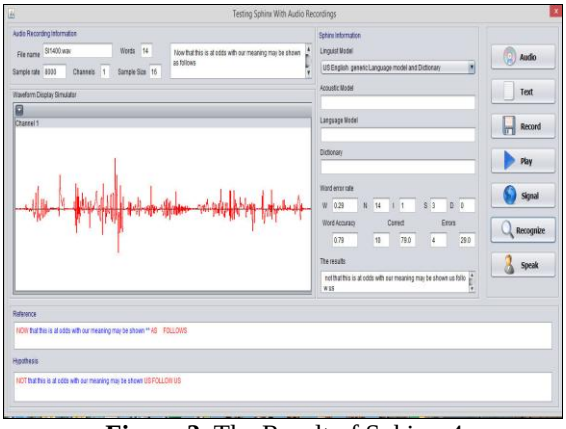

Figure 3. The Result of Sphinx-4

By using our tool, we have gathered data and results are as follows: The Sphinx-4 (37% WER), Google Speech API (9% WER) and Microsoft Speech API (18% WER). Where S sentences, N words, I words were inserted, D words were deleted, and S words were substituted. CW correct words, EW error words.

File	The Final Results of Sphinx-4								
	S	N	I	S	D	CW	EW	WA	WER
TSX223	1	8	1	1	0	6	2	0.88	0.25
TSX293	1	11	0	2	2	7	4	0.64	0.36
TSI1894	1	9	0	2	0	7	2	0.78	0.22
TSI1400	1	14	1	3	0	10	4	0.79	0.29
TSX188	2	6	0	4	0	2	4	0.33	0.67
TSI1628	2	12	0	4	3	5	7	0.42	0.58
TSX314	2	12	0	2	2	8	4	0.67	0.33
DIG001	3	15	0	1	0	14	1	0.93	0.07
TSX216	1	9	0	1	0	8	1	0.89	0.11
TSX209	1	7	0	1	1	5	2	0.71	0.29
TSI1584	2	13	0	6	2	5	8	0.38	0.62
TSX371	1	11	0	6	0	5	6	0.45	0.55
TSI1373	1	14	0	7	3	4	10	0.29	0.71
TSX233	1	7	1	2	0	4	3	0.71	0.43
OSE003	1	8	1	3	0	4	4	0.63	0.5
AENGM8	1	9	0	1	0	8	1	0.89	0.11
AENGF8	1	9	0	5	1	3	6	0.33	0.67
AENGF7	1	6	0	1	0	5	1	0.83	0.17
AENGM2	1	7	0	1	0	6	1	0.86	0.14
Mean									0.37

Table 3. The Final Results of Sphinx-4

File	The Final Results of Microsoft Speech API								
	S	N	I	S	D	CW	EW	WA	WER
TSX223	1	8	0	1	0	7	1	0.88	0.13
TSX293	1	11	0	0	0	11	0	1.0	0.0
TSI1894	1	9	0	2	0	7	2	0.78	0.22
TSI1400	1	14	0	0	0	14	0	1.0	0.0
TSX188	2	6	0	6	0	0	0	0.0	0.1
TSI1628	2	12	0	8	2	2	10	0.17	0.83
TSX314	2	12	0	0	0	12	0	1.0	0.0
DIG001	3	15	0	0	0	15	0	1.0	0.0
TSX216	1	9	0	0	0	15	0	1.0	0.0
TSX209	1	7	0	1	1	5	2	0.71	0.29
TSI1584	2	13	0	5	2	6	7	0.46	0.54
TSX371	1	11	0	1	0	10	1	0.91	0.09
TSI1373	1	14	0	5	1	8	6	0.57	0.43
TSX233	1	7	0	2	0	5	2	0.71	0.29
OSE003	1	8	0	3	0	5	3	0.63	0.38
AENGM8	1	9	0	0	0	9	0	1.0	0.0
AENGF8	1	9	0	0	0	9	0	1.0	0.0
AENGF7	1	6	0	0	0	6	0	1.0	0.0
AENGM2	1	7	0	1	0	6	1	0.86	0.14
Mean									0.18

Table 4. The Final Results of Microsoft API

File	The Final Results of Google Speech API								
	S	N	I	S	D	CW	EW	WA	WER
TSX223	1	8	0	0	0	9	0	1.0	0.0
TSX293	1	11	0	1	1	9	2	0.82	0.18
TSI1894	1	9	0	0	0	9	0	1.0	0.0
TSI1400	1	14	0	1	0	13	1	0.93	0.07
TSX188	2	6	0	0	0	6	0	1.0	0.0
TSI1628	2	12	0	2	0	10	2	0.83	0.17
TSX314	2	12	0	0	0	12	0	1.0	0.0
DIG001	3	15	0	0	0	15	0	1.0	0.0
TSX216	1	9	0	0	0	9	0	1.0	0.0
TSX209	1	7	0	0	0	7	0	1.0	0.0
TSI1584	2	13	0	5	2	6	7	0.46	0.54
TSX371	1	11	0	0	0	11	0	1.0	0.0
TSI1373	1	14	0	0	0	14	0	1.0	0.0
TSX233	1	7	1	0	0	6	1	0.71	0.14
OSE003	1	8	0	2	1	5	3	0.63	0.38
AENGM8	1	9	0	0	0	9	0	1.0	0.0
AENGF8	1	9	0	2	0	7	2	0.78	0.22
AENGF7	1	6	0	0	0	6	0	1.0	0.0
AENGM2	1	7	0	0	0	7	0	1.0	0.0
Mean									0.09

Table 5. The Final Results of Google API

File	Sphinx4		Google API		Microsoft API	
	WA	WER	WA	WER	WA	WER
TSX223	0.88	0.25	1.0	0.0	0.88	0.13
TSX293	0.64	0.36	0.82	0.18	1.0	0.0
TSI1894	0.78	0.22	1.0	0.0	0.78	0.22
TSI1400	0.79	0.29	0.93	0.07	1.0	0.0
TSX188	0.33	0.67	1.0	0.0	0.0	0.1
TSI1628	0.42	0.58	0.83	0.17	0.17	0.83
TSX314	0.67	0.33	1.0	0.0	1.0	0.0
DIG001	0.93	0.07	1.0	0.0	1.0	0.0
TSX216	0.89	0.11	1.0	0.0	1.0	0.0
TSX209	0.71	0.29	1.0	0.0	0.71	0.29
TSI1584	0.38	0.62	0.46	0.54	0.46	0.54
TSX371	0.45	0.55	1.0	0.0	0.91	0.09
TSI1373	0.29	0.71	1.0	0.0	0.57	0.43
TSX233	0.71	0.43	0.71	0.14	0.71	0.29
OSE003	0.63	0.5	0.63	0.38	0.63	0.38
AENGM8	0.89	0.11	1.0	0.0	1.0	0.0
AENGF8	0.33	0.67	0.78	0.22	1.0	0.0
AENGF7	0.83	0.17	1.0	0.0	1.0	0.0
AENGM2	0.86	0.14	1.0	0.0	0.86	0.14
Mean	WER: 0.37		WER: 0.09		WER: 0.18	

Table 6. Comparison Between Three Systems

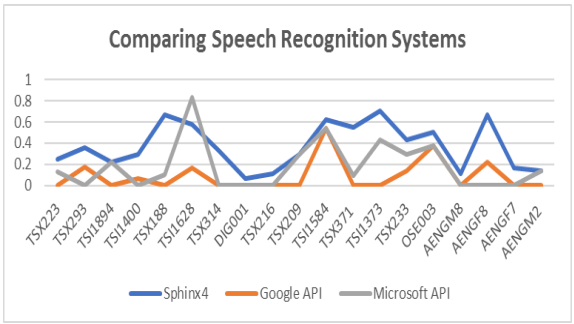

Figure 4. Comparison Between Three Systems

IX. CONCLUSION

In this paper, it can be concluded that the tool that we have built to test the Sphinx-4, Microsoft API, and Google API by using some audio recordings that were selected from many places with the original sentences showed that Sphinx-4 achieved 37% WER, Microsoft API achieved 18% WER and Google API achieved 9% WER. Therefore, it can be stated that the acoustic modeling and language model of Google is superior.

REFERENCES

[1]. W. Walker, P. Lamere, P. Kwok, B. Raj, R. Singh, E. Gouvea, P. Wolf, and J. Woelfel, Sphinx-4: A Flexible Open Source Framework for Speech Recognition, Sun Microsystems, SMLI TR-2004-139, 2004,1-14

[2]. C. Gaida, P. Lange, R. Petrick, P. Proba, A. Malatawy, and D. Suendermann-Oeft, Comparing Open-Source Speech Recognition Toolkits. The Baden-Wuerttemberg Ministry of Science and Arts as part of the research project, 2011

[3]. K. Samudravijaya and M. Barol, Comparison of Public Domain Software Tools for Speech Recognition. ISCA Archive, 2013
[4]. P. Lange and D. Suendermann, Tuning Sphinx to Outperform Google's Speech Recognition API, The Baden-Wuerttemberg Ministry of Science and Arts as part of the research project.
[5]. J. Kačur, HTK vs. Sphinx for Speech Recognition. Department of telecommunication FEI STU.
[6]. D. Isaacs and D. Mashao, A Comparison of the Network Speech Recognition and Distributed Speech Recognition Systems and their eect on Speech Enabling Mobile Devices, doctoral diss. Speech Technology and Research Group, University of Cape Town, 2010
[7]. R. Srikanth, L. Bo and J. Salsman, Automatic Pronunciation Evaluation and Mispronunciation Detection Using CMUSphin. COLING, 2012, 61-68
[8]. V. Kepuska, Wake-Up-Word Speech Recognition. IN TECH, 2011
[9]. STAR. (2016) SRI International's Speech Technology and Research (STAR) Laboratory. SRI, http://www.speech.sri.com/.
[10]. ITU. (2016) Committed to connecting the world. ITU, http://www.itu.int//.
[11]. V. Beat and J. Novet (2016) Google says its speech recognition technology now has only an 8% word error rate. Venture beat, http://venturebeat.com/2015/05/28/.
[12]. Microsoft Corporation (2016) Exploring New Speech Recognition and Synthesis APIs In Windows Vista. Microsoft, http://web.archive.org/.
[13]. CMUSphinx (2016) CMUSphinx Tutorial for Developers. Carnegie Mellon University, http://www.speech.cs.cmu.edu/sphinx/.
[14]. TIMIT (2016) TIMIT Acoustic-Phonetic Continuous Speech Corpus. Linguistic Data Consortium, https://catalog.ldc.upenn.edu/LDC93S1.
[15]. Microsoft Corporation (2016) Historic Achievement: Microsoft researchers reach human parity in conversational speech recognition", https://blogs.microsoft.com.