

Sphinx-4: Open Source Speech Recognition

Willie Walker¹, Paul Lamere¹, Philip Kwok¹
Evandro Gouvêa², Rita Singh², Bhiksha Raj³, Peter Wolf³

¹Sun Microsystems Laboratories, ²Carnegie Mellon University, ³Mitsubishi Electric Research Laboratories

<http://cmusphinx.sourceforge.net/sphinx4>

OVERVIEW

Sphinx-4 is a flexible, HMM¹-based, speaker-independent, state-of-the-art continuous speech recognition system written entirely in the Java™ programming language. Sphinx-4 is built upon a flexible, modular and pluggable framework that is designed to foster innovations in speech recognition. The framework incorporates design patterns from existing systems, with sufficient flexibility to support emerging areas of research.

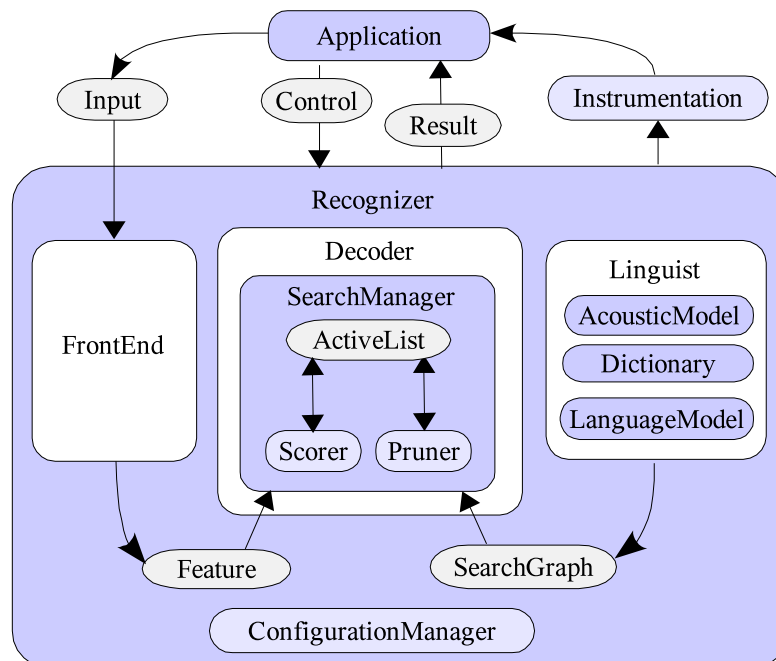
Sphinx-4 has been developed as an open source project, and is freely available under a BSD-style license on <http://cmusphinx.sourceforge.net/sphinx4>. Sphinx-4 decoder is designed jointly by researchers from Sun Microsystems Laboratories, Carnegie Mellon University (CMU) and Mitsubishi Electric Research Laboratories (MERL).

DESCRIPTION

Sphinx-4 includes a state-of-the-art HMM-based large-vocabulary recognizer. It incorporates several new features such as multistream decoding, a generalized search algorithm that subsumes Viterbi decoding as a special case, token stack decoding for efficient maintenance of multiple paths during search, etc. These new features illustrate how easy it is to experiment new techniques with Sphinx-4. There is a large vocabulary acoustic model for general use, as well as a model specialized for digits recognition.

ARCHITECTURE

There are three main blocks in the design of Sphinx-4: the FrontEnd, the Decoder, and the Linguist:



1 Hidden Markov model

The FrontEnd parameterizes input speech into a sequence of Features. The Sphinx-4 front end can simultaneously compute a variety of parameterized speech features, such as MFCC and PLP cepstra, for parallel decoding in later stages. Different features computed using independent sources, such as video, can also be used in the speech recognition process. The modularity of the front end allows easy extensibility to different feature types.

The Linguist uses information from the pronunciation dictionary and one or more set of acoustic model(s), to translate different types of language models (such as Java Speech Grammar Format (JSGF), n-Gram, finite state transducers (FST), etc.) into a SearchGraph.

The Decoder uses the SearchManager to perform the actual decoding on the SearchGraph and the features from the FrontEnd, generating Results. The default SearchManager performs a breadth-first Viterbi search on the SearchGraph, but a different SearchManager that implements another search algorithm can be easily plugged into the system.

Like other speech systems, Sphinx-4 has a large number of parameters, which can be configured using the ConfigurationManager. To give applications and developers the ability to track decoder statistics such as word error rate, run-time speed, and memory usage, Sphinx-4 provides the Instrumentation package. This package is also highly configurable, allowing users to perform a wide range of system analysis.

EXPERIMENTAL EVALUATION

To establish performance baselines, we use CMU's Sphinx-3.3 decoder, which is a high-performing state-of-the-art recognition engine written in C. Sphinx-3.3 was the last Sphinx recognition engine prior to Sphinx-4. The two primary metrics we gather are word error rate (WER) and real time speed (RT). For both metrics, the smaller the value the better:

TEST	WER (%)		REAL TIME		
	Sphinx-3.3	Sphinx-4	Sphinx-3.3	Sphinx-4 (1 CPU)	Sphinx-4 (2 CPU)
TI46 (11 words)	1.217	0.168	0.14	0.03	0.02
TIDIGITS (11 words)	0.661	0.549	0.16	0.07	0.05
AN4 (79 words)	1.300	1.192	0.38	0.25	0.20
RM1 (1000 words)	2.746	2.739	0.50	0.50	0.40
WSJ5K (5000 words)	7.323	7.174	1.36	1.22	0.96
HUB-4 (64000 words)	18.845	18.878	3.06	4.40	3.8

(This data was collected on a dual CPU UltraSPARC-III running at 1015 MHz with 2GB of memory.)

These above results show that Sphinx-4 either performs as well as or better than Sphinx-3.3 in both accuracy and speed.

CONCLUSION

Our experience with Sphinx-4 is showing that the Java platform is an optimal environment for creating a flexible, high-performing, HMM-based speaker-independent, state-of-the-art continuous speech recognition system.

ACKNOWLEDGMENTS

We thank Robert Sproull at Sun Microsystems Laboratories, Prof. Richard Stern at CMU, and Joe Marks at MERL, for making this team possible. We also thank Sun Microsystems Laboratories and the current management for their continued support and collaborative research funds. Rita Singh was sponsored by the Space and Naval Warfare Systems Center, San Diego, under Grant No. N66001-99-1-8905. The content of this paper does not necessarily reflect the position or the policy of the U.S. Government, and no official endorsement should be inferred.