Speech Decoding with SPHINX-4

Max Friedrich, Ahmed Saad, Liisa Vaht, Morteza Hagheshenas

Universität Hamburg, Speech Technology Lab, Summer Semester 2016

1. Introduction

SPHINX-4 [2] is an automatic speech recognition (ASR) framework written in the *Java* programming language. It is used both in research and consumer software.

On this poster, we describe SPHINX-4's architecture before reporting on two experiments that highlight specific features of the framework.

2. Architecture

SPHINX-4's main feature is a pluggable architecture which makes it flexible and easily extensible. Figure 1 shows an overview of the framework.

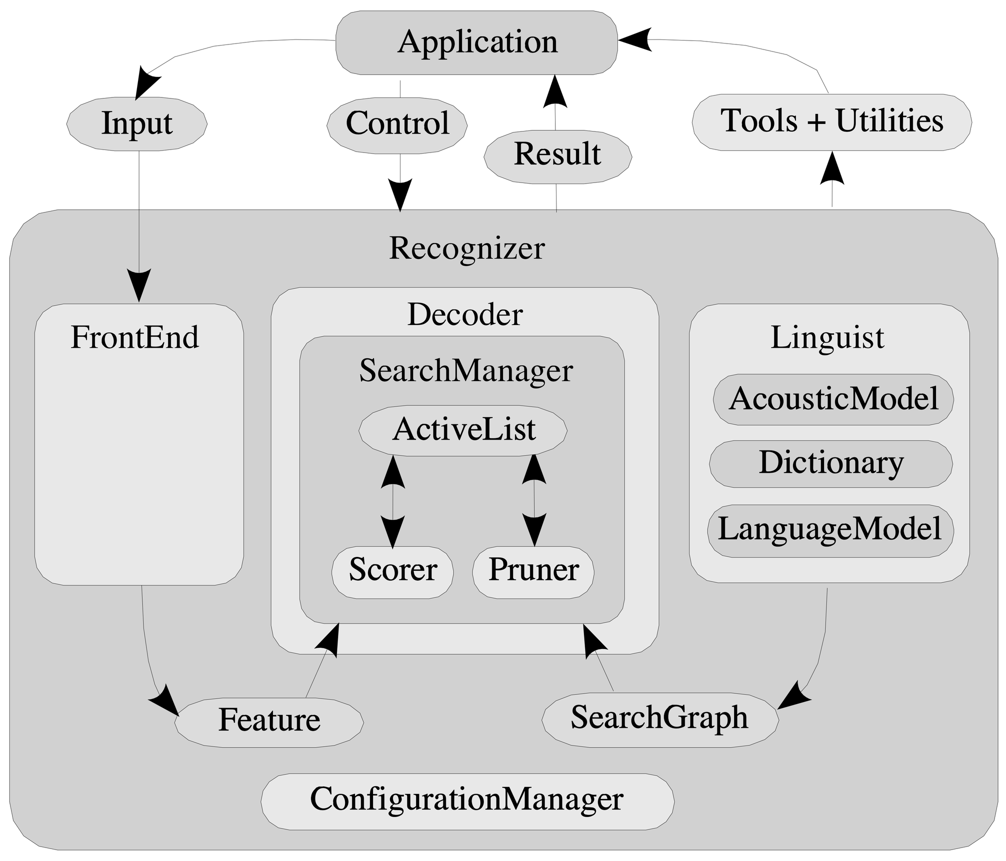

Figure 1: *Overview of the* SPHINX-4 *framework architecture. [2, Fig. 1]*

Recognizer Contains the main components of the framework. Applications interact with the SPHINX-4 system mainly via the *Recognizer*.

FrontEnd Transforms an audio signal into a sequence of features, e.g. phones.

Linguist Constructs a *SearchGraph* using a *LanguageModel*, *Dictionary*, and *AcousticModel*.

- ***LanguageModel*** Provides language structure at the word level. Typical language model implementations are either grammar-based (e.g. word list, context-free grammar) or stochastic (e.g. *n*-gram).
- ***Dictionary*** Maps words to their pronunciations, e.g. phones.
- ***AcousticModel*** Maps phones to hidden Markov models (HMMs) that can be scored against *FrontEnd* features. Training acoustic models requires very large amounts of speech data.
- ***SearchGraph*** Search space data structure, shown in Figure 2.

Decoder Combines *FrontEnd* output features and the *Linguist* output *SearchGraph* to generate speech recognition results.

SPHINX-4 provides multiple implementations of most components. They can be configured via XML files or by adapting a *ConfigurationManager* at runtime.

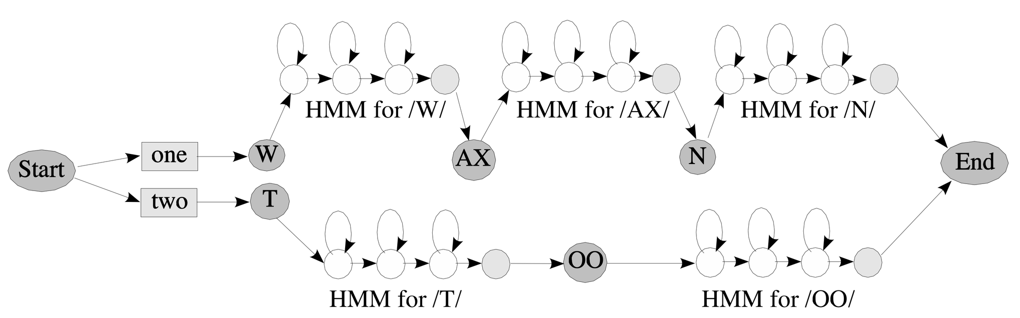

Figure 2: *An example SearchGraph [2, Fig. 3]. Contains components from LanguageModel (words in rectangles), Dictionary (sub-word units in dark circles), and AcousticModel (HMMs).*

3. Experiments

We describe two experiments that each highlight a feature of the SPHINX-4 framework. Speech recognition performance is measured by the common metric *word error rate (WER)*.

3.1 Language Models

- Compare ASR performance on connected digit speech data when using:
  1. a general purpose language model: the default stochastic `en-us.lm`.
  2. a domain-specific language model: grammar-based with only ten words (*zero* – *nine*, *oh*).

3.1.1 Setup

- 50 utterances (average length 4.86 digits) taken from the *ICSI Meeting Recorder Digits Corpus (MRD)* [1], a collection of desktop microphone recordings of various speakers made in meeting rooms.
- Recordings contain considerable noise and echo, speech volume is low.
- Normalize volume of all recordings in a preprocessing step, as no speech at all was recognized before.

3.1.2 Results

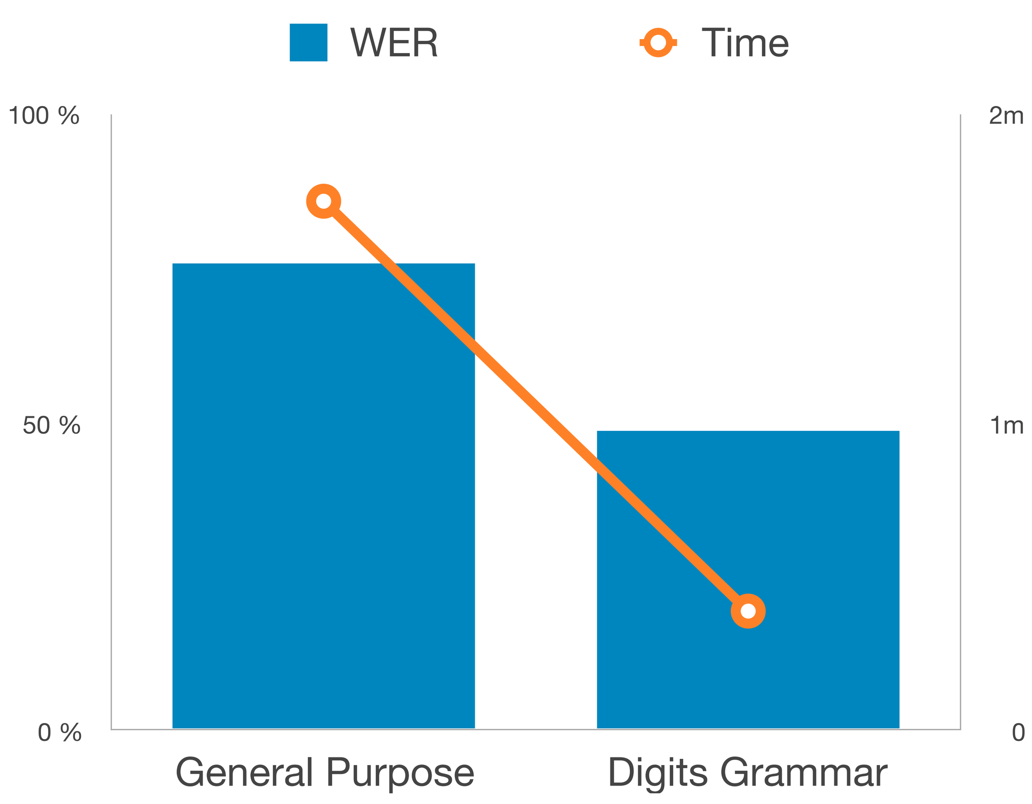

Figure 3: *Word error rate and computation time needed when transcribing 50 MRD utterances.*

3.1.3 Discussion

- Most errors originate from words not being recognized at all rather than mis-recognized, e.g. of the uttered digits "8986", the grammar recognizer transcribed only "nine eight", resulting in a 50% WER.
- Performing manual noise reduction on a small subset of the recordings improves recognition performance but reliable automatic noise reduction is out of scope for this experiment.
- Relative improvement in WER and time shows that it is reasonable to switch the language model if the expected utterance is of a specific nature, e.g. commands, "yes" / "no", or digits.

3.2 Acoustic Model Adaptation

- SPHINX-4 offers an easy way to adapt the default acoustic model to speakers, recording environment, and accents with relatively little speech data needed [3].
- Compare ASR performance on spoken text when using:
  1. a general purpose acoustic model.
  2. an acoustic model adapted to utterances of the same speaker.

3.2.1 Setup

- Spoken Wikipedia [4] article "2005 Atlantic hurricane season"
- The default acoustic model `en-us` is adapted to 50, then 150 utterances of the same speaker's recording of the article "2006 Atlantic hurricane season" (average length 8.68 words, 4.1 seconds).
- Transcribe 100 utterances of the "2005" recording (average length 9.76 words).

3.2.2 Results

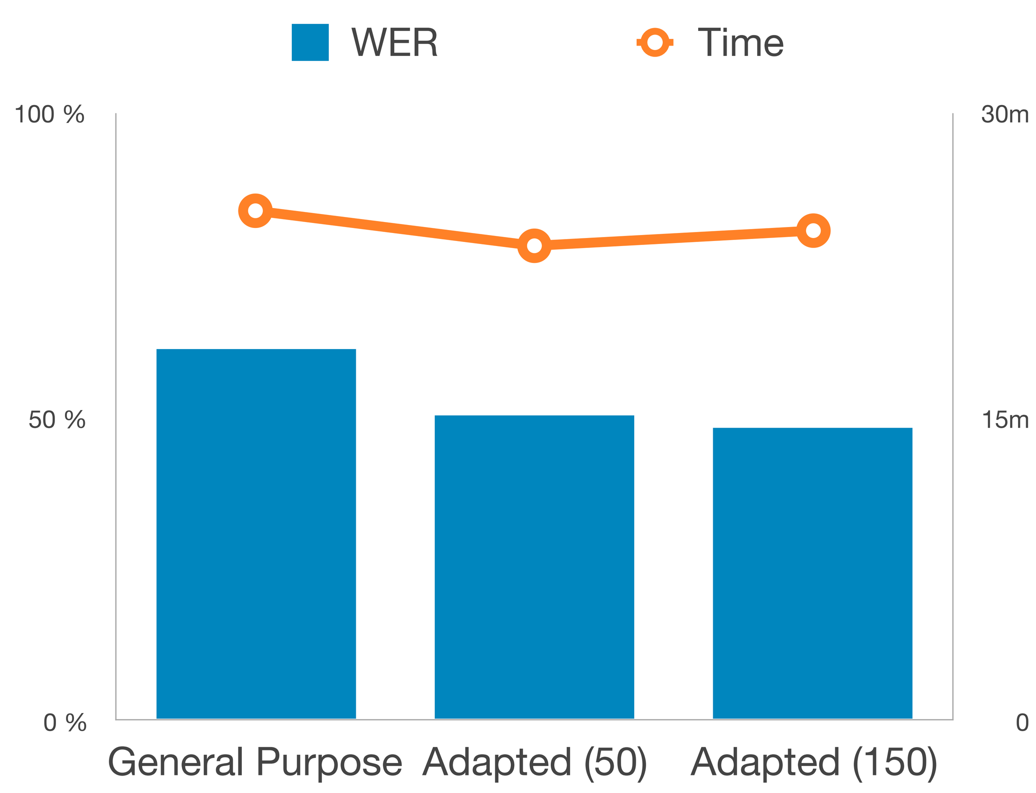

Figure 4: *Word error rate and computation time needed when transcribing 100 spoken Wikipedia utterances.*

3.2.3 Discussion

- Time required for adaptation is negligible.
- Adaptation with 50 utterances (about 200 seconds of speech data) already shows a modest improvement in WER.
- Adapted acoustic models have a significant effect on results. Of the 100 utterances examined in the "2005" article, only 11 were transcribed equally across the different models. For example, "becoming the third" was transcribed as:
  - "you coming to herd" (133% WER)
  - "becoming hard" (66% WER)
  - "becoming a third" (33% WER)
- Time needed for transcription is constant.

4. Conclusion

- SPHINX-4's architecture allows flexible configuration at runtime.
- Using a domain-specific language model yields significant WER and computation time improvements, if applicable.
- Adapting the default acoustic model is easy and yields modest WER improvements.

References

[1] Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, et al. The ICSI meeting corpus. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, volume 1, pages I–364. IEEE, 2003.

[2] Willie Walker, Paul Lamere, Philip Kwok, Bhiksha Raj, Rita Singh, Evandro Gouvea, Peter Wolf, and Joe Woelfel. Sphinx-4: A flexible open source framework for speech recognition. 2004.

[3] Sphinx Wiki. Adapting the default acoustic model. http://cmusphinx.sourceforge.net/wiki/tutorialadapt, 2016.

[4] Wikipedia. WikiProject Spoken Wikipedia. https://en.wikipedia.org/wiki/Wikipedia:Spoken_Wikipedia, 2016.