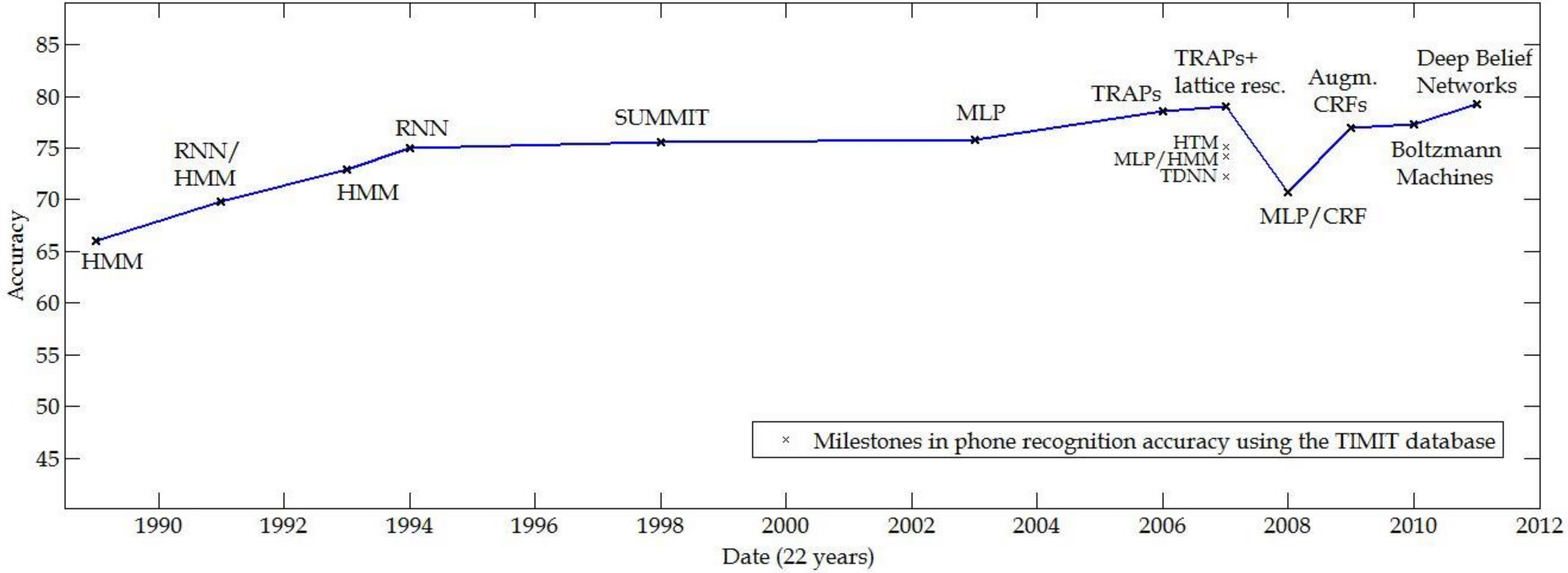
# Speech Recognition in Java

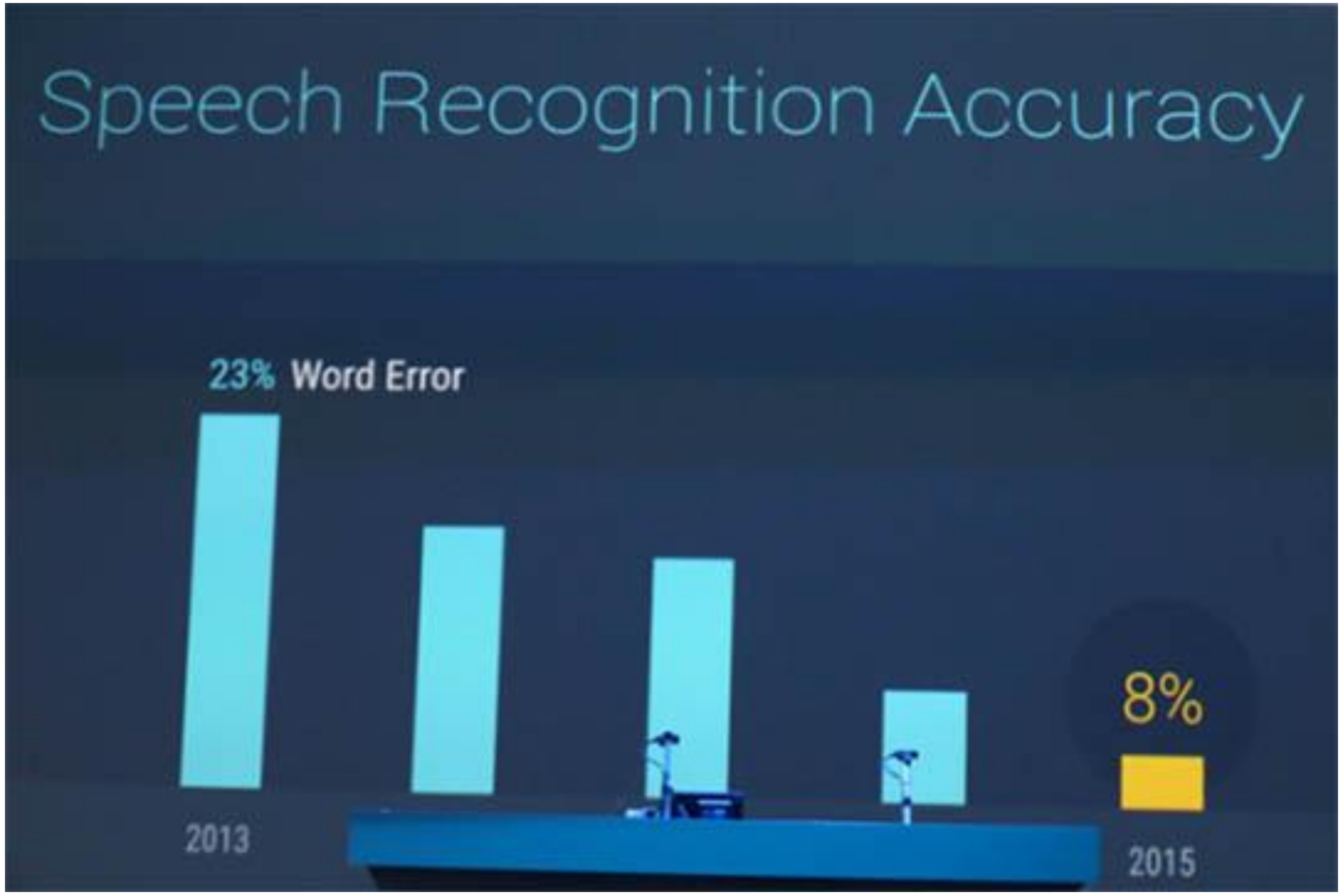
Breandan Considine

JetBrains, Inc.

# Automatic speech recognition in 2011

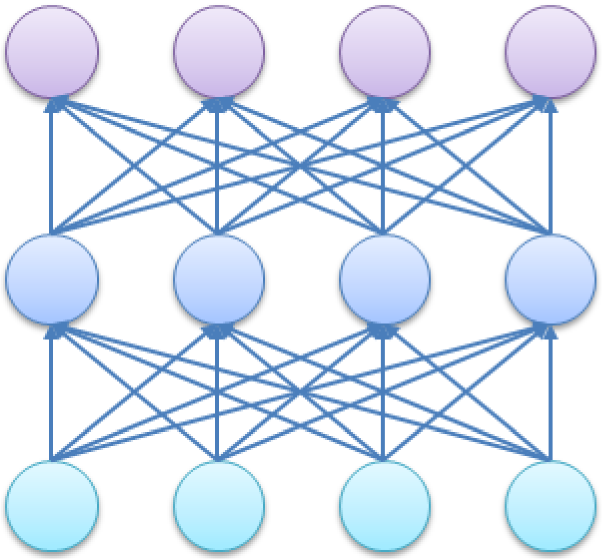


# Automatic speech recognition in 2015



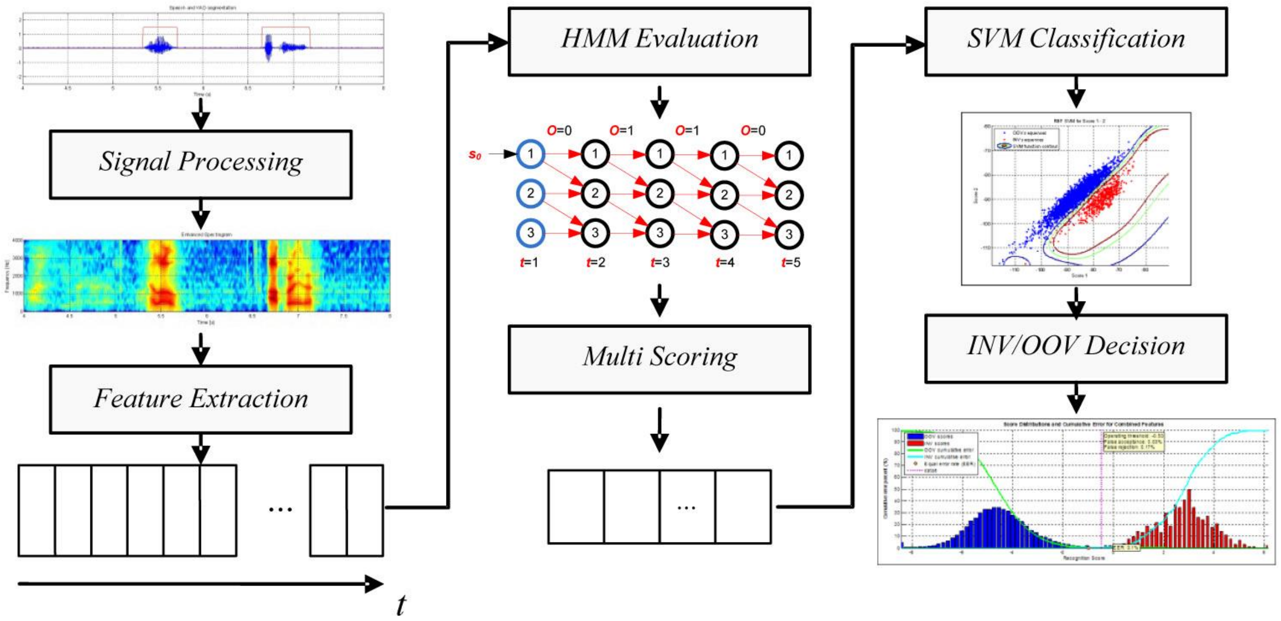
# What happened?

- Bigger data
- Faster hardware
- Smarter algorithms



# Traditional ASR

- Requires lots of handmade feature engineering
- Poor results: >25% WER for HMM architectures



# State of the art ASR

- <10% average word error on large datasets
- DNNs: DBNs, CNNs, RBMs, LSTM
- Thousands of hours of transcribed speech
- Rapidly evolving field
- Takes time (days) and energy (kWh) to train
- Difficult to customize without prior experience

# Free / open source

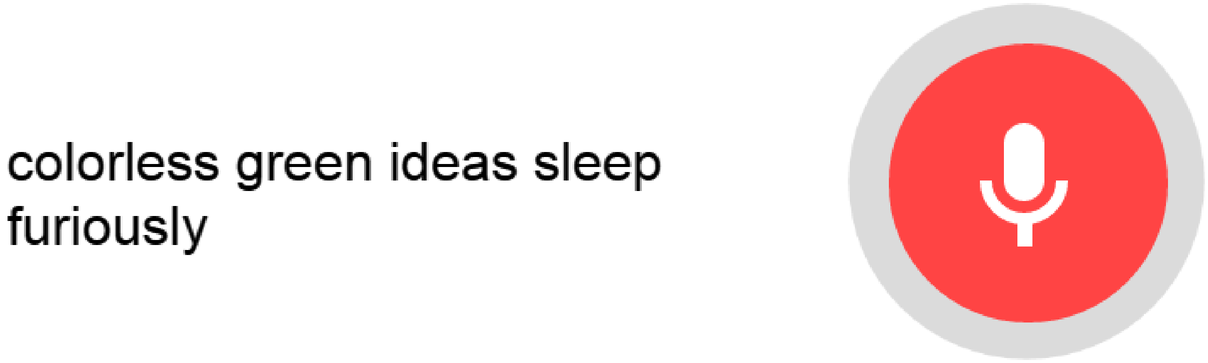
- Deep learning libraries
  - C/C++: Caffe, Kaldi
  - Python: Theano, Caffe
  - Lua: Torch
  - Java: dl4j, H2O
- Open source datasets
  - LibriSpeech – 1000 hours of LibriVox audiobooks
- Experience is required

# Let’s think…

- What if speech recognition were perfect?
  - Models are still black boxes
- ASR is just a fancy input method
- How can ASR improve user productivity?
- What are the user’s expectations?
  - Behavior is predictable/deterministic
  - Control interface is simple/obvious
  - Recognition is fast and accurate

# Why offline?

- Latency – many applications need fast local recognition
- Mobility – users do not always have an internet connection
- Privacy – data is recorded and analyzed completely offline
- Flexibility – configurable API, language, vocabulary, grammar

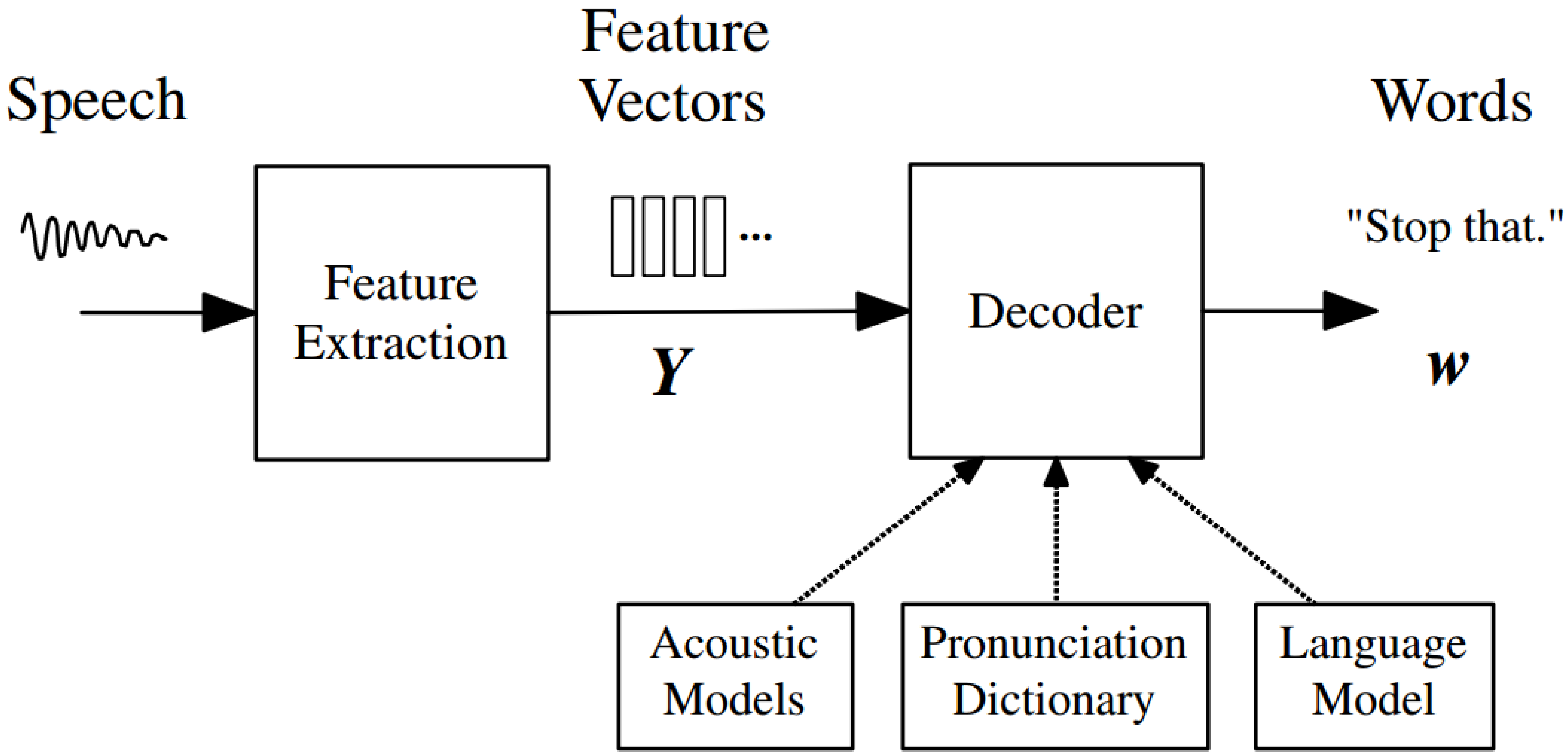


# Introduction

- What techniques do modern ASR systems use?
- How do I build a speech recognition application?
- Is speech recognition accessible for developers?
- What libraries and frameworks exist for speech?

# Maven Dependencies

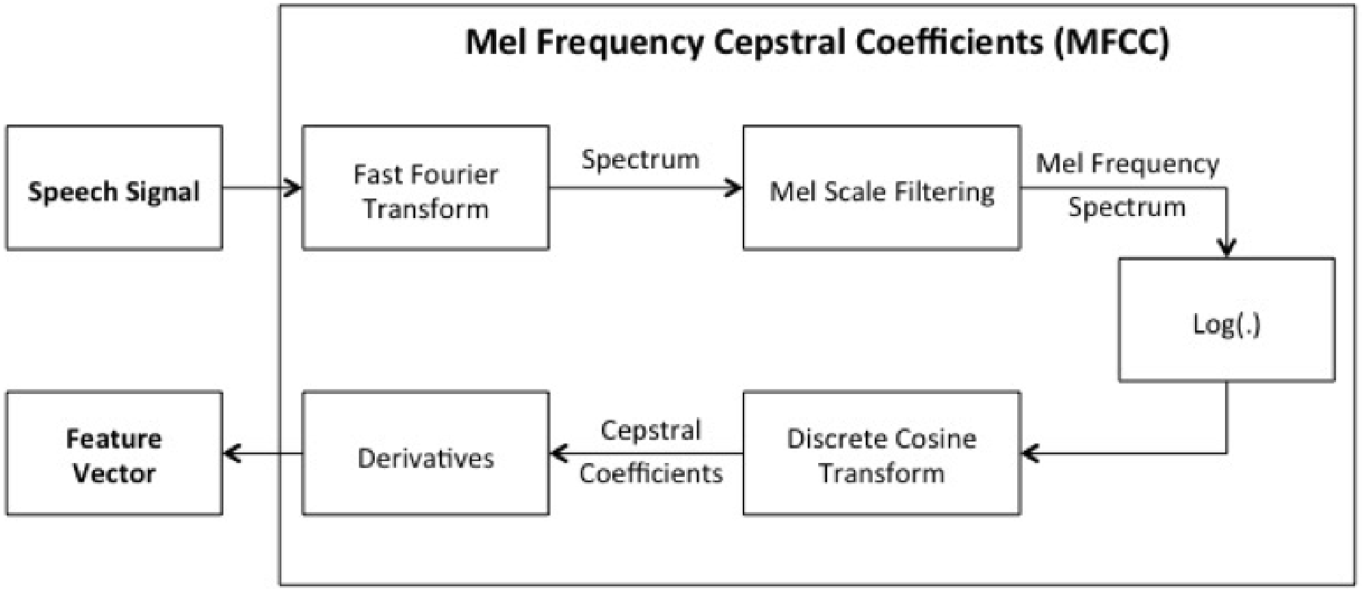
```xml
<dependency>
    <groupId>edu.cmu.sphinx</groupId>
    <artifactId>sphinx4-core</artifactId>
    <version>1.0-SNAPSHOT</version>
</dependency>

<dependency>
<groupId>edu.cmu.sphinx</groupId>
<artifactId>sphinx4-data</artifactId>
<version>1.0-SNAPSHOT</version>
</dependency>
```



Speech
Feature
Vectors
Words
Feature
Extraction
Decoder
"Stop that."
Y
w
Acoustic
Models
Pronunciation
Dictionary
Language
Model

# Feature Extraction

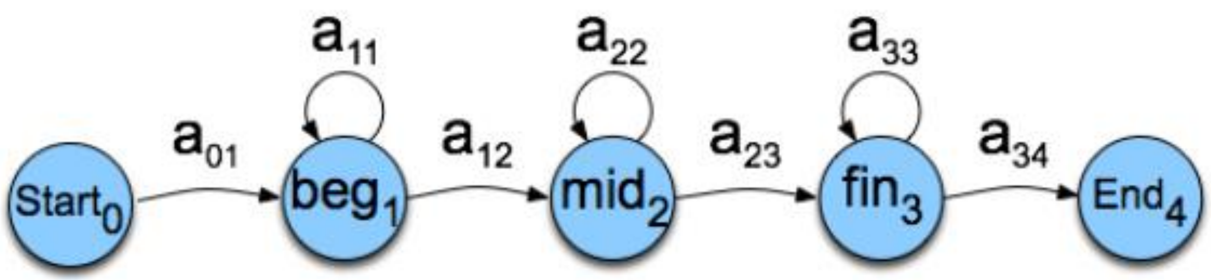
- Recording in 16kHz, 16-bit depth, mono, single channel
- 16,000 samples per second at 16-bit depth = 32KBps



# Modeling Speech: Acoustic Model

- Acoustic model training is *very* time consuming (months)
- Pretrained models are available for many languages

```
config.setAcousticModelPath("resource:<directory>");
```



# CMU Sphinx

Speech Recognition Toolkit

Brought to you by: air, arthchan2003, awb, bhiksha, and 5 others



Looking for the latest version? **Download pocketsphinx-5prealpha-win32.zip (30.8 MB)**

Home / Acoustic and Language Models

| Name | Modified | Size | Downloads / Week |
|---|---|---|---|
| Parent folder | | | |
| Kazakh | 2015-10-17 | | 213 |
| French | 2015-08-10 | | 55 |
| Russian | 2015-03-29 | | 44 |
| US English Generic Acoustic ... | 2015-02-25 | | 335 |
| Spanish Voxforge | 2015-01-19 | | 31 |
| Archive | 2015-01-15 | | 40 |
| French Language Model | 2014-11-30 | | 80 |
| US English Generic Language... | 2014-04-05 | | 117 |
| German Voxforge | 2012-07-11 | | 31 |
| Dutch Voxforge | 2012-07-11 | | 10 |
| Mexican Spanish Broadcast N... | 2012-07-11 | | 8 |
| Mandarin Language Model | 2012-07-11 | | 47 |
| Mandarin Broadcast News aco... | 2012-07-11 | | 21 |

# Modeling Text: Phonetic Dictionary

- Mapping phonemes to words
- Word error rate increases with size
- Pronunciation aided by g2p labeling
- CMU Sphinx has tools to generate dictionaries

```
config.setDictionaryPath("resource:<language>.dict");
```

# Modeling Text: Phonetic Dictionary

```
autonomous AO T AA N AH M AH S
autonomously AO T AA N OW M AH S L IY
autonomy AO T AA N AH M IY
autonomy(2) AH T AA N AH M IY
autopacific AO T OW P AH S IH F IH K
autopart AO T OW P AA R T
autoparts AO T OW P AA R T S
autopilot AO T OW P AY L AH T
```

# How to train your own language model

- Language model training is easy™ (~100,000 sentences)
- Some tools:
  - Boilerpipe (HTML text exraction)
  - Logios (model generation)
  - lmtool (CMU Sphinx)
  - IRSLM
  - MITLM

# Language model

<s> generally cloudy today with scattered outbreaks of rain and drizzle persistent and heavy at times </s>

<s> some dry intervals also with hazy sunshine especially in eastern parts in the morning </s>

<s> highest temperatures nine to thirteen Celsius in a light or moderate mainly east south east breeze </s>

<s> cloudy damp and misty today with spells of rain and drizzle in most places much of this rain will be

light and patchy but heavier rain may develop in the west later </s>

# Sphinx Knowledge Base Tool -- VERSION 3

| |
|---|
| **This is the new version of the lmtool!** **FAQ**<br><br>**Changes should be transparent (unless you automate, see note below).**<br>**Problems? Please help by sending a report to the maintainer.** |
| **New!** Follow us on `@CMUSpeechGroup` for announcements and status updates. |

**What it does:** Builds a consistent set of lexical and language modeling files for Sphinx (and compatible) decoders.

**To use:** Create a sentence corpus file, consisting of all sentences you would like the decoder to recognize. The sentences should be one to a line (but do not need to have standard punctuation). You may not need to exhaustively list all possible sentences: the decoder will allow fragments to recombine into new sentences.

**Upload a sentence corpus file**:

Choose File No file chosen

COMPILE KNOWLEDGE BASE

The **new version of lmtool** has been reorganized internally to make use of the Logios package. This will make lmtool easier to maintain in the future and will allow it to take advantage of ongoing development in Logios. These changes should be transparent to regular users. Please give it a try. If you have any problems, or discover bugs, let the maintainer know. If things look good (i.e., I stop getting bug reports) this will become the standard version.

**NOTE:** If you have automated the use of this tool you will need to update your code. The main difference is that the name of the target script has changed. The old script will still be available so nothing will break immediately, but it's unlikely to continue to be maintained. Also, file links are no longer tagged in the html. Please let me know if you make use of this feature and I'll find a fix.

*Alex Rudnicky*

# Sphinx knowledge base generator [lmtool.3a]

---

Your Sphinx knowledge base compilation has been successfully processed!

The base name for this set is **4072**. TAR4072.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 1445903127_06784
[_INFO_] Found corpus: 5 sentences, 77 unique words
[_INFO_] Found 0 words in extras  (0)
[_INFO_] Language model completed  (0)
[_INFO_] Pronounce completed  (0)
[_STAT_] Elapsed time: 1.768 sec
```

Please include these messages in bug reports.

| Name | Size | Description |
|---|---|---|
| 4072.dic | 1.2K | *Pronunciation Dictionary* |
| 4072.lm | 5.7K | *Language Model* |
| 4072.log_pronounce | 802 | *Log File* |
| 4072.sent | 493 | *Corpus (processed)* |
| 4072.vocab | 353 | *Word List* |
| TAR4072.tgz | 2.8K | COMPRESSED TARBALL |

---

# Modeling Speech: Grammar Model

- JSpeech Grammar Format

```
config.setGrammarPath("resource:<grammar>.gram");
```

```
<size> = /10/ small | /2/ medium | /1/ large;
<color> = /0.5/ red | /0.1/ blue | /0.2/ green;
<action> = please (/20/save files |/1/delete files);
<place> = /20/ <city> | /5/ <country>;
public command = <size> | <color> | <action> | <place>
```

# Modeling Speech: Grammar Format

```
public <number> = <hundreds> | <tens> | <teens> | <ones>;
      <hundreds> = <ones> hundred
                   [<tens> | <teens> | <ones>];
          <tens> = ( twenty | thirty | forty | fifty |
                     sixty | seventy | eighty | ninety )
                   [<ones>];
         <teens> = ten | eleven | twelve | thirteen |
                   fourteen | fifteen | sixteen |
                   seventeen | eighteen | nineteen;
          <ones> = one | two | three | four | five | six |
                   seven | eight | nine;
```

# Configuring Sphinx-4

```
Configuration config = new Configuration();

config.setAcousticModelPath(AM_PATH);
config.setDictionaryPath(DICT_PATH);
config.setLanguageModelPath(LM_PATH);
config.setGrammarPath(GRAMMAR_PATH);
// config.setSampleRate(8000);
```

# Live Speech Recognizer

```
LiveSpeechRecognizer recognizer =
    new LiveSpeechRecognizer(config);

recognizer.startRecognition(true);
…
recognizer.stopRecognition();
```

# Live Speech Recognizer

```
while (…) {
    // This blocks on a recognition result
    SpeechResult sr = recognizer.getResult();

    String h = sr.getHypothesis();
    Collection<String> hs = sr.getNbest(3);
    …
}
```

# Stream Speech Recognizer

```
StreamSpeechRecognizer recognizer = new
StreamSpeechRecognizer(configuration);
recognizer.startRecognition(
        new FileInputStream("speech.wav"));
SpeechResult result = recognizer.getResult();
recognizer.stopRecognition();
```

# Improving recognition accuracy

- Using context-dependent cues
- Structuring commands to reduce phonetic similarity
- Disabling the recognizer
- Grammar swapping
- Busy waiting

# Grammar Swapping

```
static void swapGrammar(String newGrammarName) throws
PropertyException, InstantiationException, IOException
{
    Linguist linguist = (Linguist)
                        cm.lookup("flatLinguist");
    linguist.deallocate();
    cm.setProperty("jsgfGrammar", "grammarName",
                        newGrammarName);
    linguist.allocate();
}
```

# MaryTTS: Initializing

```
maryTTS = new LocalMaryInterface();
Locale systemLocale = Locale.getDefault();
if (maryTTS.getAvailableLocales()
      .contains(systemLocale)) {
    voice = Voice.getDefaultVoice(systemLocale);
}

maryTTS.setLocale(voice.getLocale());
maryTTS.setVoice(voice.getName());
```

# MaryTTS: Generating Speech

```
try {
    AudioInputStream audio = mary.generateAudio(text);
    AudioPlayer player = new AudioPlayer(audio);
    player.start();
    player.join();
} catch (SynthesisException | InterruptedException e)
{
    …
}
```

# Resources

- CMUSphinx, http://cmusphinx.sourceforge.net/wiki/
- Deep Learning for Java, http://deeplearning4j.org/
- MaryTTS, http://mary.dfki.de/
- FreeTTS 1.2, http://freetts.sourceforge.net/
- JSpeech Grammar Format, http://www.w3.org/TR/jsgf/
- ARPA format for N-gram backoff (Doug Paul) http://www.speech.sri.com/projects/srilm/manpages/ngram-format.5.html
- Language Model Tool http://www.speech.cs.cmu.edu/tools/lmtool.html

# Further Research

- Accurate and Compact Large Vocabulary Speech Recognition on Mobile Devices, research.google.com/pubs/archive/41176.pdf
- Comparing Open-Source Speech Recognition Toolkits, http://suendermann.com/su/pdf/oasis2014.pdf
- Tuning Sphinx to Outperform Google's Speech Recognition API, http://suendermann.com/su/pdf/essv2014.pdf
- Deep Neural Networks for Acoustic Modeling in Speech Recognition, research.google.com/pubs/archive/38131.pdf
- Deep Speech: Scaling up end-to-end speech recognition, http://arxiv.org/pdf/1412.5567v2.pdf

# Special Thanks

- Alexey Kudinkin (@alexeykudinkin)
- Yaroslav Lepenkin (@lepenkinya)
- CMU Sphinx (@cmuspeechgroup)
- JetBrains (@JetBrains)
- Hadi Hariri (@hhariri)