Kaldi project for ZRE

Open source speech recognition

Karel Vesely

Speech@FIT, BUT

Brno, 3.5.2016

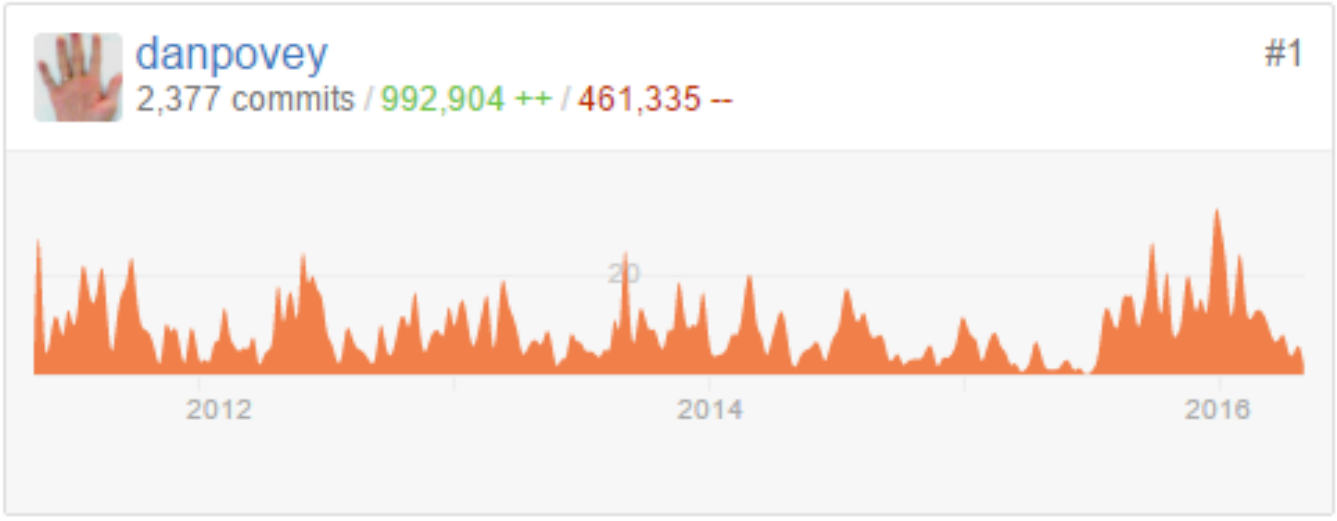
What is Kaldi?

- Wiki: A legendary Ethiopian goatherd who discovered the coffee plant.
- Github: Open-source toolkit for building **speech recognition** systems.

A bit of history...

- 2009: Summer workshop at Johns Hopkins University (Baltimore, USA)
  - ASR team worked on *Sub-space Gaussian Mixture Models* (part of model parameters is shared across languages)
  - A toolkit was needed to integrate the new model!
- 2010: Dan Povey started coding Kaldi at Microsoft
- **2010, 2011, 2012, 2013: Kaldi development workshops**
  - **Every year an international team of self-funded volunteers gathered in Brno for several weeks of summer coding in 'zámeček' at FIT**
- 2011: Kaldi toolkit presented at conferences ICASSP (Prague), ASRU (Hawaii)
- 2012: Dan Povey joins JHU (leaving Microsoft)
- 2015: Kaldi moved from SourceForge to **GitHub**

Who is this 'Dan Povey'?



- The '#1', i.e. the main architect of Kaldi.
- He is believed to write C++ code at the speed of light!

What is Kaldi? II.

Kaldi = GitHub project[1], it consists of:

- Set of command-line **programs for training and representing speech recognition models** (C++).
- **example recipes** = set of **"standard experiments"** on cluster computer (BASH, perl, awk, SGE cluster)
- **Documentation[2]:** Doxygen with tutorial, topic-based pages and C++ code reference
- **Support:** discussion forum, email

[1] https://github.com/kaldi-asr/kaldi

[2] http://kaldi-asr.org/doc/

The example recipes = main strength of Kaldi

The recipes are main strength of Kaldi compared to other toolkits!
(HTK, Sphinx, Julius, ...)

- Toy examples: yes/no, tidigits,
- Free-databases: AMI meetings (80h), TED-LIUM talks (120h), voxforge,
- The standard tasks (from easy to difficult):
  - **Read speech:** Resource Management (3h, WER=1.5%), TIMIT (3h), Wall Street Journal (80h, **WER**[3]**=4%**),
  - **Conversational telephone speech:** Switchboard (300h, **WER=10%**), Fisher (2000h)
  - **Spontaneous "distant multi-microphone" speech:** AMI meetings (80h*8, distant multi-mic **WER=36%**, headset-mic **WER=23%**)

[3] WER = word error rate

Why is Kaldi good for research?

- **Experiments are very easy to reproduce:**
  (all researchers can work with same baseline system)
- No need to implement everything from scratch
- The toolkit is easy to extend or modify
- It is a **community project**:
  - Anybody can propose a change
  - send bugfix
  - fork and create derived project
- **License:** Apache v2.0, liberal formulation: The toolkit can be modified, used commercially, or parts can be reused.

Speech recognition research ecosystem

Researchers:
- are using the toolkit
- some are contributors

Big companies:
- some use Kaldi
- all have access to the code

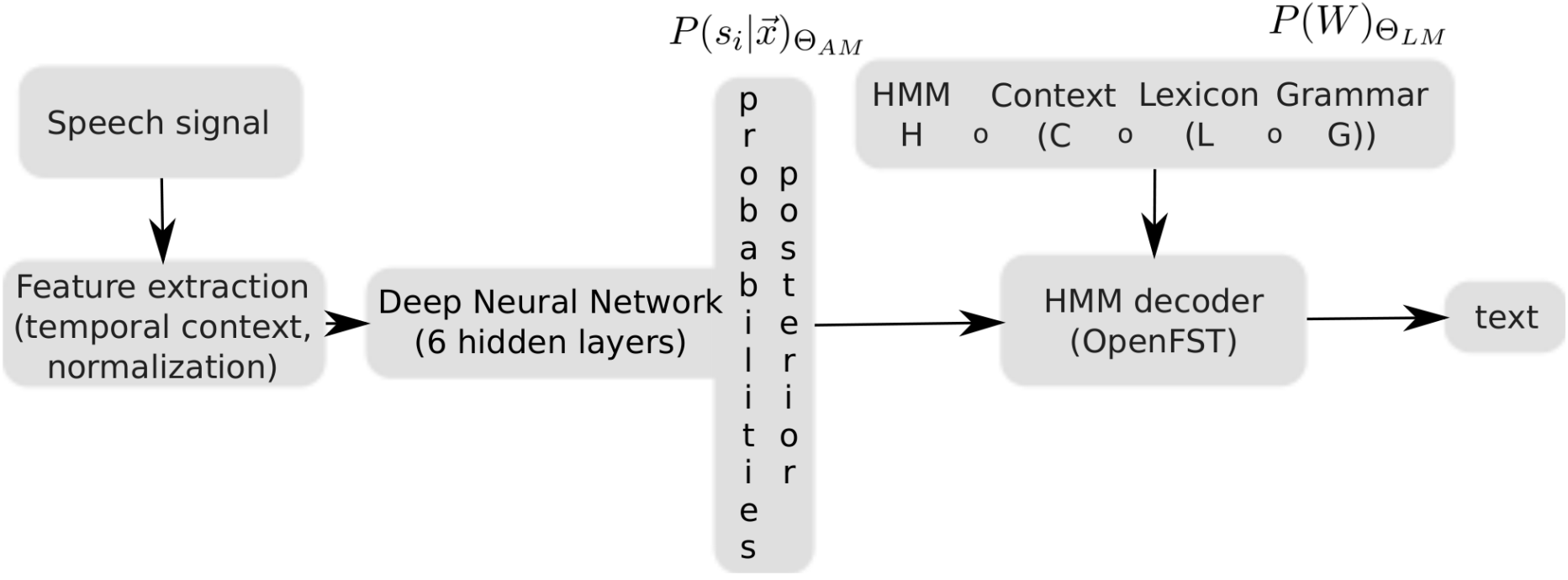
Start-ups:
- getting free ASR technology
- creating new ASR applications

Big companies doing speech research: Nuance, IBM, Google, Microsoft, Apple, Amazon, Baidu, Telefonica, Samsung.
Many have open work positions...

Implemented techniques

- Speech recognition:
  - HMM decoder using WFST transducers
  - keyword search based on WFSTs
  - Acoustic models: GMM, SGMM, DNN (nnet1,2,3)
    (DNN types: feed-forward, Convolutional, LSTM, BLSTM)
  - Language models: N-GRAM, RNNLM
- Speaker identification
- Language identification

Speech recognition: A hybrid approach



The decoding formula:

$$\tilde{W} = \underset{W}{\operatorname{argmax}} P(W|X)_{\Theta} \propto \underset{W}{\operatorname{argmax}} P(X|\vec{s}_W)_{\Theta_{AM}} P(W)_{\Theta_{LM}}$$

We use Bayes rule to convert NN posteriors into likelihoods:

$$P(\vec{x}|s_i)_{\Theta_{AM}} = P(s_i|\vec{x})_{\Theta_{AM}} / P(s_i)$$

Training the DNN

- supervised training of a classifier
  (input features classified into triphone tied-states),
- training labels generated from transcriptions and existing model,
- training algorithm: mini-batch Stochastic Gradient Descent:

$$\vec{w}_{t+1} = \vec{w}_t - \eta \nabla E(\vec{w}_t)$$

- avoiding over-training by reducing learning rate according to the loss of the held-out set

Advanced techniques

- speaker adaptations (CMVN, fMLLR, i-vector based)
- sequence-discriminative training bMMI, sMBR
- multi-GPU DNN training with Natural Gradient pre-conditioning
- nnet1 is supporting serio-parallel structures
- nnet3 is supporting generic graph structures

Where we use Kaldi

- In research, for publishing results in conference articles,
- For cooperation with international colleagues,
- In research projects with funding,

What can you do with Kaldi

- Play with toy examples, tidigits.
- Think of a creative application, where speech recognition would be used
  (pre-built models are available `http://kaldi-asr.org/downloads/all/`).

Useful links

GitHub project:

- **https://github.com/kaldi-asr/kaldi**

Documentation:

- **http://kaldi-asr.org/doc/**

Support forum:

- https://groups.google.com/forum/#!forum/kaldi-help

Other resources:

- http://www.danielpovey.com/kaldi-lectures.html
- http://www.danielpovey.com/publications.html
- http://www.danielpovey.com/
- http://kaldi-asr.org

The end

Thank you!