Free on-line speech recogniser based on Kaldi ASR toolkit producing word posterior lattices

Ondřej Plátek and Filip Jurčíček {oplatek, jurcicek}@ufal.mff.cuni.cz
Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University in Prague

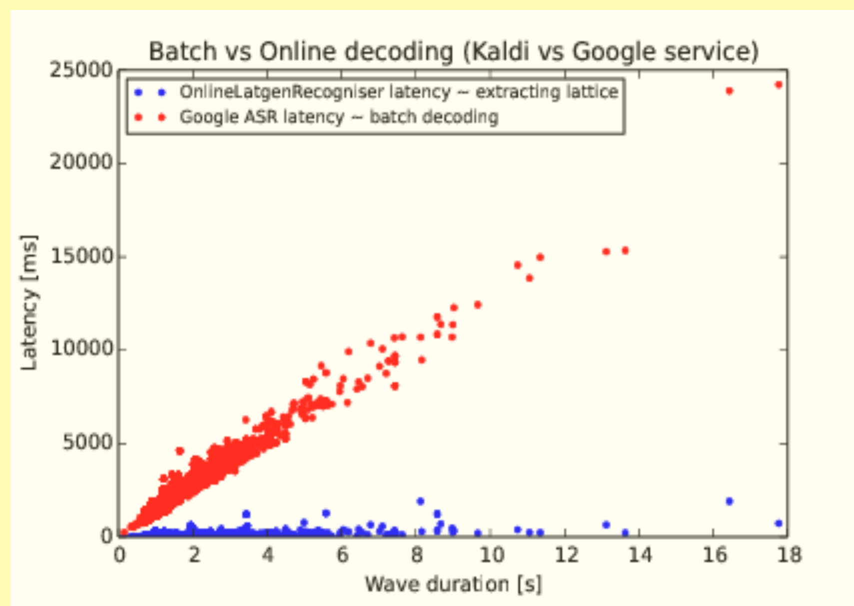
Motivation: Batch versus Online Decoding

Batch Decoding

- Waits for the end of the utterance to start decoding
- Latency increases linearly with the utterance length

Online Decoding

- Incremental processing in small chunks
- Result: **low latency**



The Kaldi ASR Toolkit

- Based on Finite-State Transducers
- State-of-the-art acoustic modelling techniques
- Well maintained by an enthusiastic community
- Fast enough
- Lacked support for online decoding

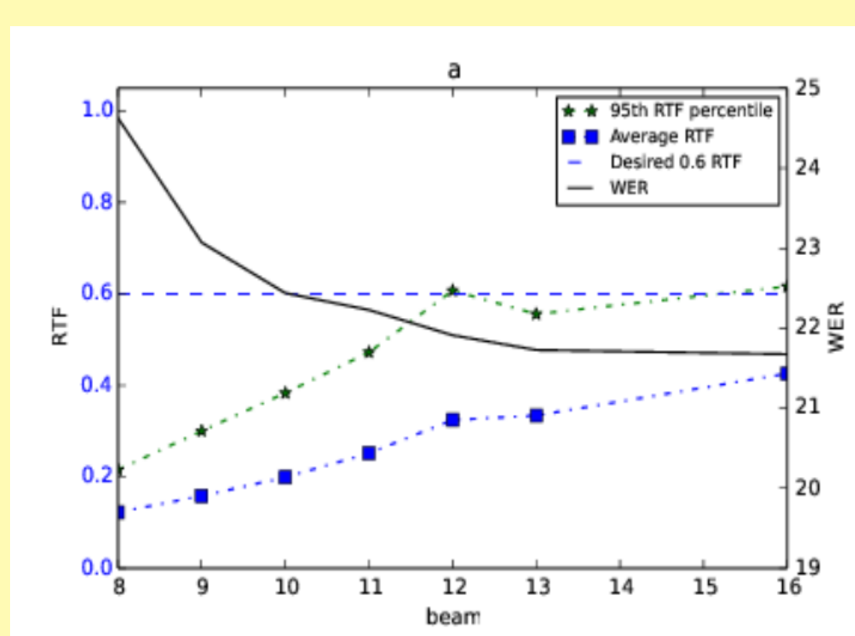
Motivation: **Get Kaldi's high performance with low latency for use in a Spoken Dialogue System**

Evaluation in a Spoken Dialogue System

- Tested in production environment in the Alex spoken dialogue system framework
  - Czech public transport information domain

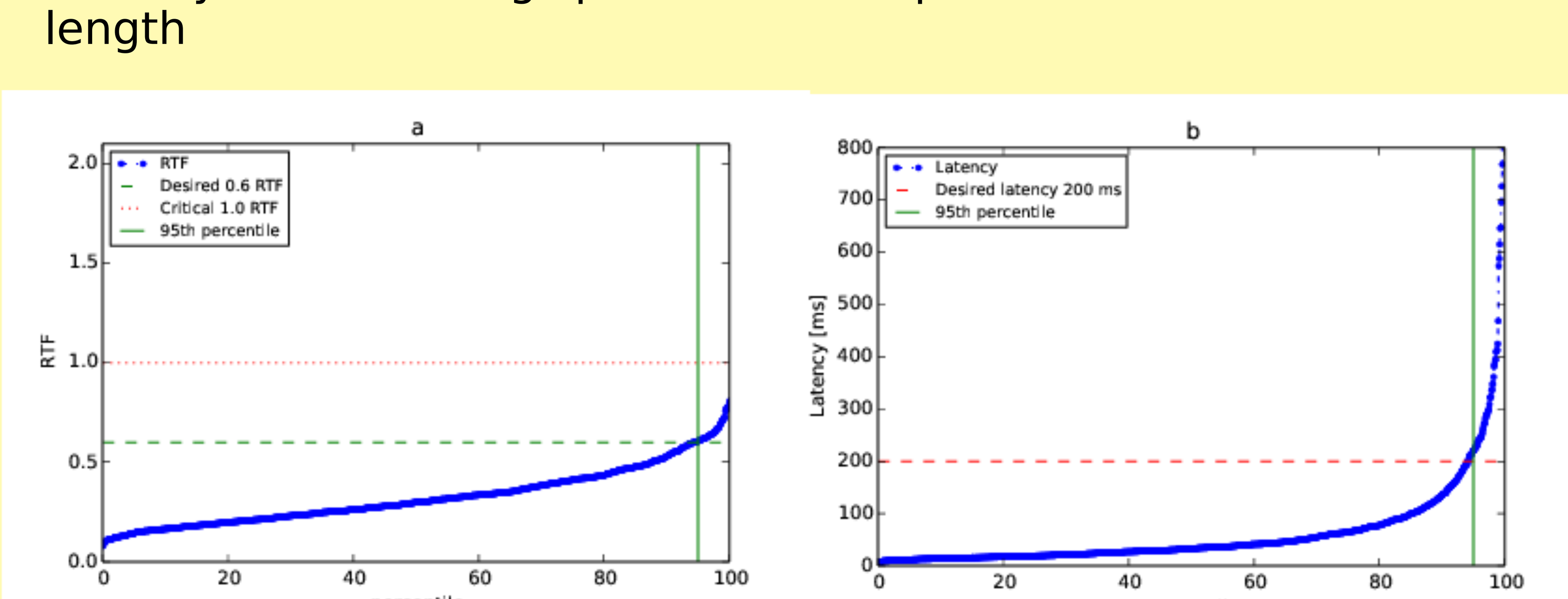
Parameter grid search

- ***beam*** – controls a dynamic number of alternative ASR hypotheses
- ***max-active-states*** – a threshold on the number of alternative hypotheses
- ***lattice-beam*** – level of approximation during phoneme lattice extraction



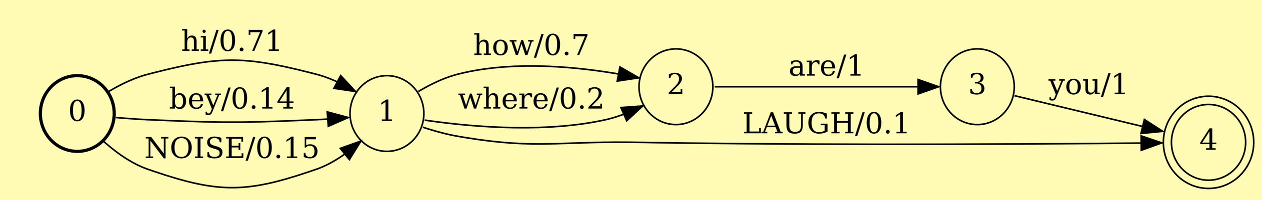
Evaluation

- On 1000 recorded utterances from the Alex system, from previously unseen dialogues
- Utterance length varies
- **WER: 22%**
- Decoder **latency well below 200 ms in 95% cases**
- Noisy utterances slow down the decoder
- Latency and decoding speed do not depend on utterance length



OnlineLatgenRecognizer Design

- Simple and responsive
- Robust
- Guaranteed latency
- Iterative decoding
- Supports LDA + MLLT, bMMI, MPE
- Straightforward C++ interface
- Python extension
- Outputs Word Posterior Lattices



C++ API

AudioIn(audio)
- Accepts audio.

Decode(max_frames)
- Decodes at most max_frames

PruneFinal()
- prepares decoder for lattice extraction.

GetLattice()
- extracts lattice

Reset()
- prepare for new utterance

GetBestPath()
- single output

```
OnlineLatgenRecogniser rec;
rec.Setup(...);
size_t decoded_now = 0, max_decode = 10;
char *audio_array = NULL;

while (recognitionOn()){
  size_t audio_len = getAudio(audio_array);
  rec.AudioIn(audio_array, audio_len);
  do {
    decoded_now = rec.Decode(max_decode);
  } while(decoded_now > 0);
}
rec.PruneFinal();
double tot_lik;
fst::VectorFst<fst::LogArc> word_post_lat;
rec.GetLattice(&word_post_lat, &tot_lik);
rec.Reset();
```

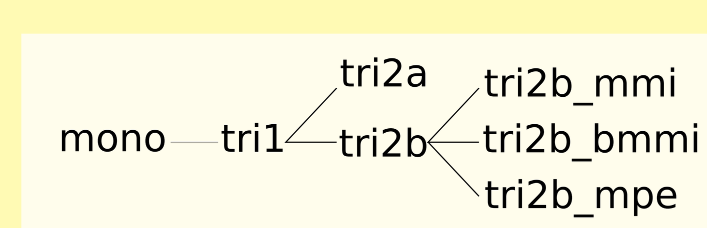
Thin Python Wrapper

```
class AsrSimplifiedInAlex:
  def rec_in(self, frame):
    self.decoder.frame_in(frame.payload)
    dec_t = self.decoder.decode(max_frames)
    while dec_t > 0:
      frame_total += dec_t
      dec_t = self.decoder.decode(max_frames)

  def hyp_out(self):
    self.decoder.prune_final()
    utt_lik, lat = self.decoder.get_lattice()
```

Training Scripts for Acoustic Modelling

- **Speaker-independent models** for Kaldi
- **LDA+MLLT+bMMI**
- Advanced acoustic models retrained **based on simpler models**, monophones trained from flat start



Training Data Sizes

dataset	audio[hour]	# sentences	# words
English			
training	41:30	47,463	178,110
development	01:45	2,000	7,376
test	01:46	2,000	7,772
Czech			
training	15:25	22,567	126,333
development	01:23	2,000	11,478
test	01:22	2,000	11,204

Results

Method	bigram WER
tri Δ + ΔΔ	56.6
tri LDA+MLLT	53.9
tri LDA+MLLT+MMI	49.5
tri LDA+MLLT+bMMI	49.3
tri LDA+MLLT+MPE	49.2

Method	bigram WER
tri Δ + ΔΔ	16.2
tri LDA+MLLT	15.8
tri LDA+MLLT+MMI	10.4
tri LDA+MLLT+bMMI	10.2
tri LDA+MLLT+MPE	11.1

Summary

- **Apache 2.0 license** ✓
- **Simple C++ API, easy to use Python thin wrapper** ✓
- **Used in the Alex spoken dialogue system** ✓
- **High quality word posterior lattices** ✓
- **Training scripts for free acoustic data** ✓

This research was funded by the Czech Ministry of Education under the grant agreement LK11221 and by core research funding of Charles University in Prague. Language resources presented in this work are stored and distributed by the LINDAT/CLARIN project of the Czech Ministry of Education (project LM2010013).