Abkhazia – ASR experiments made easy

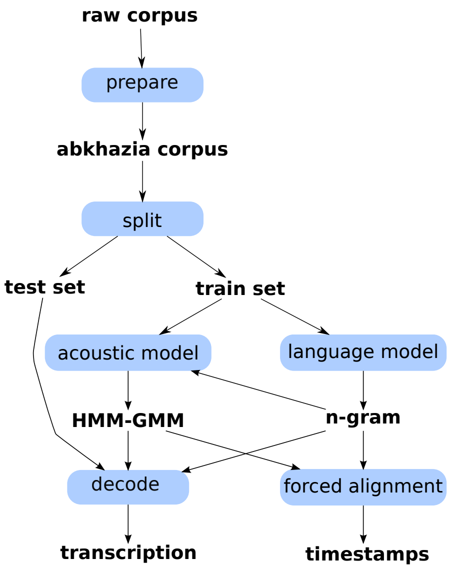
Xuan Nga Cao   Mathieu Bernard

LSCP Bootphon Team Meeting – March 24, 2016

Abkhazia – ASR experiments made easy

- Abkhazia is a Python library and a **command-line tool**
  - sources and installation instructions at *https://github.com/bootphon/abkhazia*
  - free software (GPL3)
  - in development (about 70% done). . .
- Performs **various ASR tasks**
  - language and acoustic models, forced-alignment, decoding
  - use the Kaldi toolkit (kaldi-asr.org)
  - local or parallel computation (on the cluster)
  - uniform command-line syntax
- Defines and rely on a **standard speech corpus format**
  - inspired by Kaldi format
  - support for several corpora (WSJ, Buckeye, etc. . .)
  - possible extension to new corpora

The big picture – what abkhazia can do



Outline

Standard speech corpus format

TODO XN

Supported corpora

Actually, abkhazia have preparation scripts for:

- **aic** - Articulation Index Corpus LSCP
- **buckeye** - Buckeye Corpus of conversational speech
- **csj** - Corpus of Spontaneous Japanese
- **globalphone** - GlobalPhone multilingual read speech corpus
  - Mandarin, Vietnamese
- **librispeech** - LibriSpeech ASR Corpus
- **wsj** - Wall Street Journal ASR Corpus
- **xitsonga** - NCHLT Xitsonga Speech Corpus

Abkhazia commands

- ASR tasks are spread over abkhazia *commands*:
  - **prepare** - prepare a speech corpus for use with abkhazia
  - **split** - split a corpus in train and test subsets
  - **language** - compute a language model
  - **train** - train (or retrain) an acoustic model
  - **decode** - compute phone posteriograms or transcription
  - **align** - compute forced-aligment
- All commands share some basic functionalities
  - parameters from command-line and/or configuration file
  - logging system, help messages

abkhazia prepare: [raw] -> [corpus]

Convert a corpus from its raw distribution to the abkhazia format

- **input**: any supported raw corpus
- **output**: abkhazia corpus in `<output>/data`
- **options**: corpus specific
- **examples**:
  - `abkhazia prepare wsj --help`
  - `abkhazia prepare buckeye -v -i ./raw_buckeye`
  - `abkhazia prepare globalphone -j 4 -l vietnamese`

abkhazia split: [corpus] -> [corpus], [corpus]

Split a corpus in train and test subsets

- **input**: any abkhazia corpus
- **output**: test and train sets in
  - `<output>/train/data`
  - `<output>/test/data`
- **key options**:
  - `--test-prop`: proportion of utterances in test set
  - `--by-speaker`: split by speaker (default is by utterance)
- **examples**:
  - `abkhazia split --help`
  - `abkhazia split --by-speaker ./input_corpus`
  - `abkhazia split -t 0.25 -r 0 buckeye`

abkhazia language: [corpus] -> [lm]

Compute a *n*-gram language model from an abkhazia corpus

- **input**: any abkhazia corpus
- **output**: `language_model.fst` and a kaldi recipe dir
- **key options**:
  - `--model-order`: n in n-grams
  - `--model-level`: phone or word language model
- **examples**:
  - `abkhazia language --help`
  - `abkhazia language --model-level word -v buckeye`
  - `abkhazia language --model-order 3 ./input_corpus`

abkhazia train: [corpus], [lm] -> [model]

Train a standard speaker-adapted triphone HMM-GMM model from a prepared corpus. Write the directory `<corpus>/model`.

abkhazia align: [model], [lm] -> [result]

Generate a forced-alignment from acoustic and language models.
Write the directory `<corpus>/align`.

abkhazia decode: [corpus], [model], [lm] -> [result]

Decode a prepared corpus from a HMM-GMM model and a language model. Write the directory `<corpus>/decode`.

Example – forced-alignment of a buckeye subset

TODO XN

Conclusion

- And the answer is. . .
- $f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n$