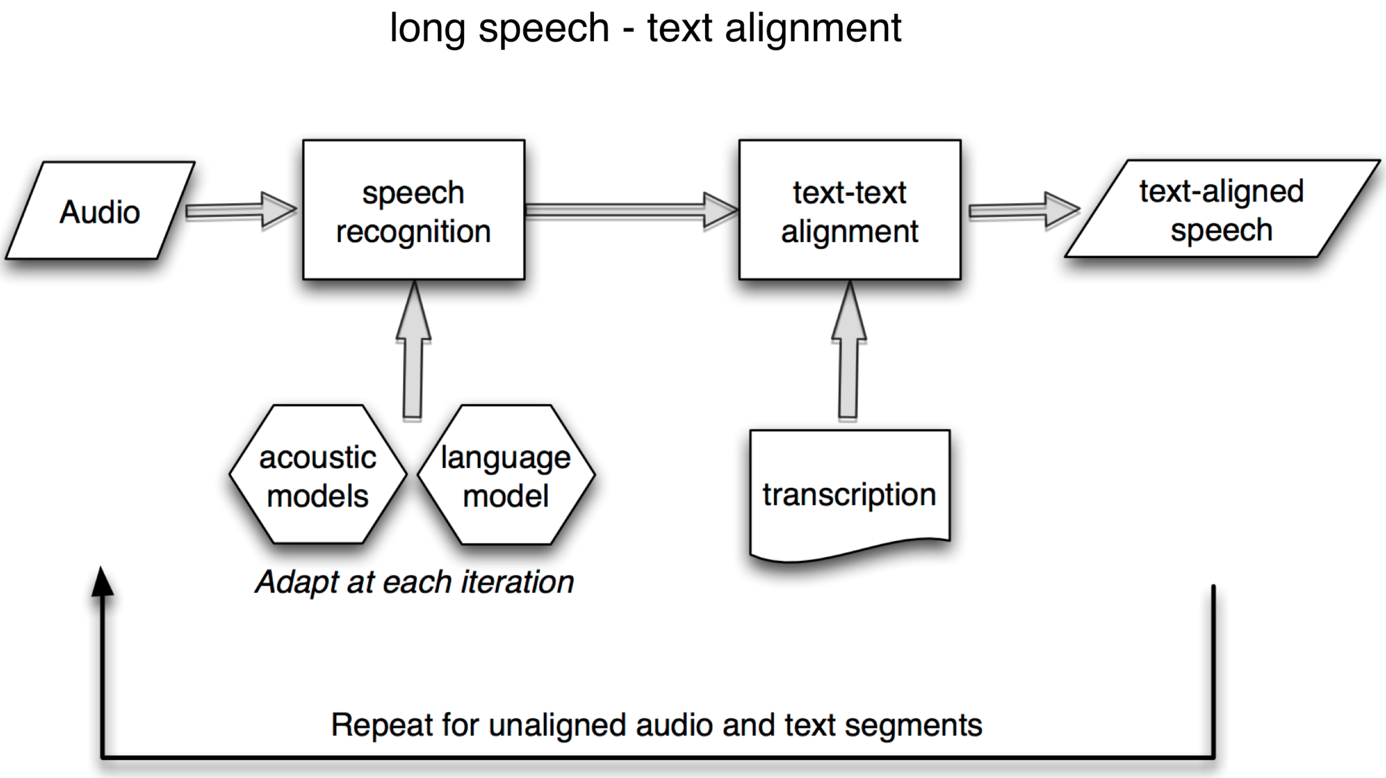
SailAlign Tutorial

Nassos Katsamanis
Signal Analysis and Interpretation Laboratory (SAIL)
University of Southern California
http://sipi.usc.edu/~nkatsam

robust long speech-text alignment

- Conventional Viterbi-based forced alignment suffers from the following limitations:
  - The transcription has to be accurate
  - Audio has to be relatively noise-free
- SailAlign circumvents these restrictions by implementing an alignment scheme which builds upon the iterative segmental application of a large vocabulary continuous speech recognition system, that was introduced by Moreno et al. in ICSLP '98.



A. Katsamanis, M. Black, P. Georgiou, L. Goldstein and S. Narayanan,
SailAlign: Robust long speech-text alignment,
in Proc. of Workshop on New Tools and Methods for Very-Large Scale Phonetics Research, Jan. 2011.

SailAlign is freely distributed: http://sail.usc.edu/software/SailAlign

the architecture

- SailTools
  - Library of perl packages, implementing the overall speech-text alignment scheme
  - Has been designed to allow integration of custom speech recognition, acoustic adaptation and language modeling tools
  - Currently SailAlign has only been tested with HTK, SRILM, SCTK.
- Hidden Markov Model Toolkit (HTK)
  - HDecode and HVite are used as speech recognition engines.
  - HERest is used for the adaptation of the acoustic models.
  - HCopy is used for acoustic feature extraction from audio.
- SRI Language Modeling Toolkit (SRILM)
  - ngram and ngram-count are used for language model building.
- SCTK
  - sclite is used for text-text alignment.

usage

```
>> sail_align –i support/data/timit_5.wav –t support/data/timit_5.txt
              –w support/data/test/timit_sample_test –e timit_sample_test
              –c config/timit_alignment.cfg
```

switch	corresponding option
-i	audio file (wav format, 16kHz, 16bits, 1 channel)
-t	text file with the audio transcription in one line
-w	working directory where output files will be stored
-e	an experiment id for easy reference to a specific alignment run
-c	configuration file

the transcription

support/data/timit_5.txt:

tradition requires parental approval for under age marriage don't ask me to carry an oily rag like that it's healthier to cook without sugar few rural areas are protected by zoning rector was often curious often tempted to ask questions but he never did mom strongly dislikes appetizers the cranberry bog gets very pretty in autumn brown eyes eyebrow mustache she had your dark suit in greasy wash water all year pizzerias are convenient for a quick lunch don't ask me to carry an oily rag like that nowadays we talk as though the blitz were just a short skirmish elderly people are often excluded her wardrobe consists of only skirts and blouses drop five forms in the box before you go out put the butcher block table in the garage jars are assembled in bowl of butter mold propriety was synonymous with ritual observance the mark of a true gentleman she had your dark suit in greasy wash water all year don't ask me to carry an oily rag like that a flame would use up air shaving cream is a popular item on halloween they own a big house in the remote countryside help greg to pick a peck of potatoes his history is his alone yet each man must recognize his own history in it michael colored the bedroom wall with crayons every movement she made seemed unnecessarily noisy the misquote was retracted with an apology planned parenthood organizations promote birth control she had your dark suit in greasy wash water all year a boring novel is a superb sleeping pill…

- No punctuation
- All text in one line

alignment results

support/test/ref/timit_sample_test/timit_5.lab:

```
0.71 1.12 TRADITION
1.12 1.6 REQUIRES
1.6 1.98 PARENTAL
1.98 2.42 APPROVAL
2.42 2.58 FOR
2.58 2.87 UNDER
2.87 3.56 AGE MARRIAGE
3.91 4.08 DON'T
4.08 4.31 ASK
…
```

- Each line corresponds to an aligned segment.
- Start and end times are in seconds
- The .lab file is compatible with transcription browsing software, e.g., Wavesurfer.

configuration

config/timit_alignment.cfg

```
%cfg    = (
…
alignment       => {
    do_adaptation                   => 1,   # Adapt acoustic models at each iteration
    do_phon_alignment               => 1,   # Do forced phonetic alignment in the end
    do_forced_word_alignment        => 1,   # Do forced word alignment in the end
    min_n_aligned_words             => 4,   # Min number of words for reliably aligned region
    acoustic_model                  => {
        defs => catdir(
                $ROOTPATH,
                'models/ac_models/english/htk/wsj_si84_2750_8/hmmdefs'
                ), …
```

forced alignment & the .wrd file

In the configuration file:

```
…
 word_forced_alignment      => {
        …
        utt_duration => 600,      # Duration in seconds for each segment
                                  # to be forced aligned
       tool        => 'htk',
         …
},

…
```

the .phn file & .wrd files

support/test/ref/timit_sample_test/timit_5.forced.phn:

```
0.71 0.73 sil
0.73 0.77 t
0.77 0.8 r
0.8 0.83 ah
0.83 0.88 d
0.88 0.95 ih
0.95 1.06 sh
1.06 1.09 ih
1.09 1.12 n
1.12 1.12 sp
…
```

- The result of the word-level forced alignment is in the .wrd file and is in the same format as the .lab file.

the dictionary (1)

In the configuration file (**config/timit_alignment.cfg**):

```
 dictionary => {
           tool           => 'htk',
          reference       => [
               'language/cmu_dictionary.dic' ,
               'language/timit_dictionary.dic'
           ],
          apply_phone_map => 1,
          phone_map_direct => 'language/timit2cmu_phones.map',
          phone_map_inverse => 'language/cmu2timit_phones.map',
          file     => catfile( $WORKINGDIR, 'dictionary' ),
        },
```

the dictionary (2)

language/cmu_dictionary.dic

```
ACADEMIA         ae k ah d iy m iy ah
ACADEMIC         ae k ah d eh m ih k
ACADEMICALLY  ae k ah d eh m ih k l iy
ACADEMICIAN   ae k ah d ah m ih sh ah
```

language/timit_dictionary.dic

```
AMBIGUITY ae m b ax g y uw ix t iy
AMBIGUOUS ae m b ih g y uw ax s
AMBITIONS ae m b ih sh ix n z
AMBITIOUS ae m b ih sh ix s
```

language/timit2cmu_phones.map

```
eng ih
ax  ah
ix  ih
el  l
em  m
en  n
axr er
```

porting to a new language/phoneset (1)

- There are two alternative ways to do that:
  - a) Find a mapping between the new phoneset and the one that is currently used
    - Easy to implement
      1. Provide the new dictionary (in the .cfg file)
         - » dictionary => {
           reference => [‘language/new_language_dictionary.dic’],}
      2. Provide the direct and inverse mappings between the phonesets (in the .cfg file)
         - » phone_map_direct => 'language/new2cmu_phones.map’,
         - » phone_map_inverse => 'language/cmu2new_phones.map’,
    - SailAlign, has been tested for the conversion from the timit phoneset to the CMU phoneset.
    - Of course, a perfect mapping may not (and usually does not) necessarily exist.

porting to a new language/phoneset (2)

- b) So, train acoustic models based on the new phoneset/ language may be a better solution.
  - This is a solution that is a bit more involved.
  - Acoustic models have to be HTK models. For further details check Ketih Vertanen's webpage and software for building HTK models, http://www.keithv.com/software/htk/ Of course, you will need transcribed acoustic data for your new language.
  - Currently, the acoustic features used for the models are 'MFCC_0_D_A_Z'. We assume that this does not change for the new models.
  - We need to provide the new dictionary. No phonetic mappings are necessary.
  - We need to provide the new acoustic model locations in the .cfg file, i.e., update all acoustic_model related structures in the file

the log file

support/test/ref/timit_sample_test/alignment.log:

```
…
DEBUG - audio:support/data/timit_5.wav text: support/data/timit_5.txt
DEBUG - Initialized experiment timit_sample_test, working dir: support/test/ref/timit_sample_test
DEBUG - Successfully initialized a SailSignal instance
DEBUG - Initialized transcription
DEBUG - Properly initialized alignment
DEBUG - Signal: support/data/timit_5.wav Number of transcribed words: 864
DEBUG - /home/work/speech_text_alignment/sail_align/bin/vad -m /home/work/speech_text_alignment/sail_align/
      models/vad/MattModel.bin -i support/data/timit_5.wav -o support/test/ref/timit_sample_test/vad/
      voice_activity.out --ST-window-size 0.02 --ST-window-shift 0.01
DEBUG - File's duration is: 301.16 sec
DEBUG - /home/work/speech_text_alignment/sail_align/bin/HCopy -T 1 -C support/test/ref/timit_sample_test/
      feature_extraction.cfg support/data/timit_5.wav support/test/ref/timit_sample_test/features/asr/htk/support/
      data/timit_5.mfc
DEBUG - Feature extraction finished OK
DEBUG - Starting feature file segmentation
…
INFO - Number of aligned words in the segment/total number of words: 11/11
INFO - Percentage of aligned words: 0.898148148148148
DEBUG - current_unaligned_index=7 ind = 3 number_of_segs = 22
…
```

SailAlign

- More information on Sunday at 13:00 and at:

http://sail.usc.edu/SailAlign

- To acquire SailAlign and for technical support, just send me an email: nkatsam@sipi.usc.edu

- Currently only working on linux
  - Has been tested on 32 and 64 bits architectures
  - There is a problem with HDecode on Mac I am trying to figure out
- Installation
  - It is installed as a perl package

```
>>perl  Build.PL
>> ./Build
>> sudo ./Build install
```