07/20/2017

**JSALT Rosetta sub-team**

Shruti, Francesco, Lucas, Mark, Florian, and Odette

**Research question 1**

Can we find the phoneme speech tokens of an under-resourced language through adaptation of an ASR system trained on a resourced language?

*Assumption 1*: we know “the” phoneme inventory of the unknown language (e.g., from Wikipedia, knowing that these are far from the “truth” (because what is the actual phoneme inventory of a language? There will always be disputes about that).

*Assumption 2*: we know (roughly) what language family the under-resourced language belongs to.

*Assumption 3*: there is a resourced language that is somewhat related to the under-resourced language (ideally from the same language family); although in principle the setup should work from any resourced language not necessarily one close to the under-resourced language.

**Methodology**

Well-resourced language: Dutch (read speech from CGN); size is XXX

Under-resourced language: English (FlickR\_8K); matched in size with Mboshi, size is XXX

1. Train a DNN-based phone recognition system (in EESEN) on Dutch → Baseline model

Token error rate = XX%

1. Compare the phone inventories of the well-resourced and under-resourced language, identify phones from the under-resourced language (= ‘L2 phones’) that are missing from the well-resourced language, identify the closest phone from the well-resourced language (= ‘L1 phone’) in terms of:
   1. phonetic/articulatory feature distance
   2. visual inspection of the distance of the L1 phone categories in the soft-max layer of the Baseline model.

Size of Dutch phone inventory: XX phones, in SAMPA

Size of English phone inventory: 39 phones, in ARPABET

Number of L2 phones that have an equivalent or closely matching L1 phone: 29

Number of L2 phones that are represented by a sequence of two L2 phones: 5. We are currently leaving these as is and will combine the two L1 phones into the English L2 phone in a post-processing step after decoding:

|  |  |  |
| --- | --- | --- |
| **Missing L2 phone** | **Example word** | **Sequence mapping of Dutch L1 phones** |
| AY | life | A + j |
| CH | choke | t + j |
| JH | cage | d + Z |
| OY | joy | O + j |
| UW | flew | y + w |

Number of missing L2 phonemes: 5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Missing L2 phone** | **Example word** | **Mapping** | **Mapping** | | |
|  |  |  | **L1:1** | **L1:2** | **L1:3** |
| AE | map | halfway between E and a: | E | E | a: |
| AH | cut | origin = E + half the distance between a: and A | E | a: | A |
| DH | they | halfway between v and z | v | v | z |
| ER | b**i**rd/church | halfway between 2 and o: | 2 | 2 | o: |
| TH | three | halfway between f and s | f | f | s |

1. Remove all Dutch phones that are not part of the English phoneme inventory from the soft-max layer.
2. To add the missing L2 phones to the soft-max layer of the system trained on Dutch materials, vectors are created for these L2 phones on the basis of the trained Dutch L1 phones. This is done by extrapolating the missing L2 vector from existing vectors for the L1 phones in the Baseline model using:

where is the vector of the missing L2 phone *φ* that needs to be created, and are the vectors of the Dutch L1 phones *φ* in het soft-max layer that are used to create the vector for the missing L2 *φ*, where *L1:1* refers to the phone which is used as the starting point from which to extrapolate the missing L2 phone, and *L1:2* and *L1:3* refer to the L1 phones whose distance is used as an approximation of the distance between the Dutch L1 vector and the to be created English L2 vector.

1. Run a free phone recognition pass on the under-resourced language. This will result in a phone alignment of the under-resourced language or in other words creates ‘self-labels’ of the under-resourced language.
2. Compare the result of the self-labelling to the ‘gold standard’ phone transcription of FlickR\_8K. This is a check to make sure that the removal of the not-in-English-existing Dutch phones does not disrupt/reconfigure the soft-max layer too much (e.g., by making previously unlikelier phones suddenly very likely). This will give us a TER.
3. Extract the ASR confidence scores for each sentence in the under-resourced language.
4. Retrain the Baseline system in a semi-supervised fashion using the sentences from FlickR\_8K with the highest confidence scores and the self-labels created during the free phone recognition pass:
   1. On all utterances/alignments in FlickR\_8K.
   2. On only those utterances/alignments with the ‘highest’ confidence score according to 7.
5. Evaluate the Baseline system (before copying the weights) and the new model after each iteration of step 5-7 (= Iter model <nb>) on the decoding task (free phone loop) of FlickR\_8K: compare the free phone loop with the ‘gold standard’ of FlickR\_8K.

**Research question 2**

Does retraining a supervised model in a different (well-resourced) language lead to discover better phone speech tokens than training a model using a semi-supervised training with self-labels?

**Methodology**

1. Train a system from scratch using only/all the utterances from the under-resourced language, where the utterances have been described using the Baseline system in the previous section.
2. Run a free phone recognition pass on the under-resourced language. This will result in a phone alignment of the under-resourced language or in other words creates ‘self-labels’ of the under-resourced language.
3. Compare the result of the self-labelling to the ‘gold standard’ phone transcription of FlickR\_8K. This will give us a TER.
4. Extract the ASR confidence scores for each sentence in the under-resourced language.
5. Retrain the system from scratch in a semi-supervised fashion using the sentences from FlickR\_8K with the highest confidence scores and the self-labels created during the free phone recognition pass:
   1. On all utterances/alignments in FlickR\_8K.
   2. On only those utterances/alignments with the ‘highest’ confidence score according to 7.
6. Possibly several iterations of steps 2-5.
7. Evaluate system on a phone-to-translated-text retrieval task.
8. Compare the newly trained system(s) to the adapt-supervised-model in the previous section.

**Research question 3**

Which confidence scores are best for selecting the best phone sequences for (re)training the system for discovering phone speech tokens?

**Methodology**

Compare

1. ASR confidence scores
2. Phone-sequence-to-image task
3. Phone-sequence-to-translated-text task
4. TTS

Evaluation, as before:

1. Comparison to the ground truth
2. Phone-sequence-to-translated-text retrieval task.

Note: This RQ can only be investigated on multi-modal (speech, pictures, translated text) databases.

**Wanted extras**, in order of wantingness:

1. Visualisation of the DNN hidden layers, mostly for RQ1:
   * 1. Which phones were correctly learned? Compare to the system trained from scratch
     2. How are the phones in the Iter model <nb> spread out compared to the Baseline model and the system trained from scratch?
     3. Compare the phone distribution of the Iter model <nb> with a monolingual model trained on the under-resourced language (supervised training, which is of course not possible in the case of an actual under-resourced language)

→ This is the reason why we want to carry out this experiment on Dutch and English, to get a feel for what to expect when we move to an actual under-resourced language for which we can’t build a monolingual supervised model

1. As ‘proper’ evaluation criterion, instead of a comparison with the ground truth (which we wouldn’t have for a low-resource language), and as a second measure to determine the confidence score, we want to use a translated text retrieval task. We are assuming that field linguists are (more) likely to have a translation of recordings (than having recordings containing descriptions of pictures). To that end, a phone-string-to-translated-text system should be build using the phone sequences created by the free phone loop as training material.
   1. Input phone sequences: English phone sequences created by the free phone loop decoding tasks
   2. Input translated text: Japanese translated sentences in text format corresponding to the test sentences.
   3. Task: retrieval of the correct translated text from the test database (note, this is a relatively small test set).
   4. Evaluation measures: precision, recall, F-score
2. Phone-string-to-image system, trained in the same way as the phone-string-to-translated-text system, used for:
   1. Evaluation of the systems, this evaluation measure assumes that the field linguists have recordings in which humans are describing pictures.
   2. As a measure to determine the confidence score which can be used as a selection criterion for iterative training steps.
3. TTS, see I and II.

**Tasks:**

* Subsampling of FlickR\_8K to match the size of Mboshi: who?
* Evaluation on translated text retrieval: Odette
  + installation of dynet etc
  + creating the right training scripts
  + running an upper baseline version with the gold standard of the English FlickR\_8K text and their Japanese translations
* Adapting the soft-max layer + free phone loop: Francesco?
* Visualisation: who?

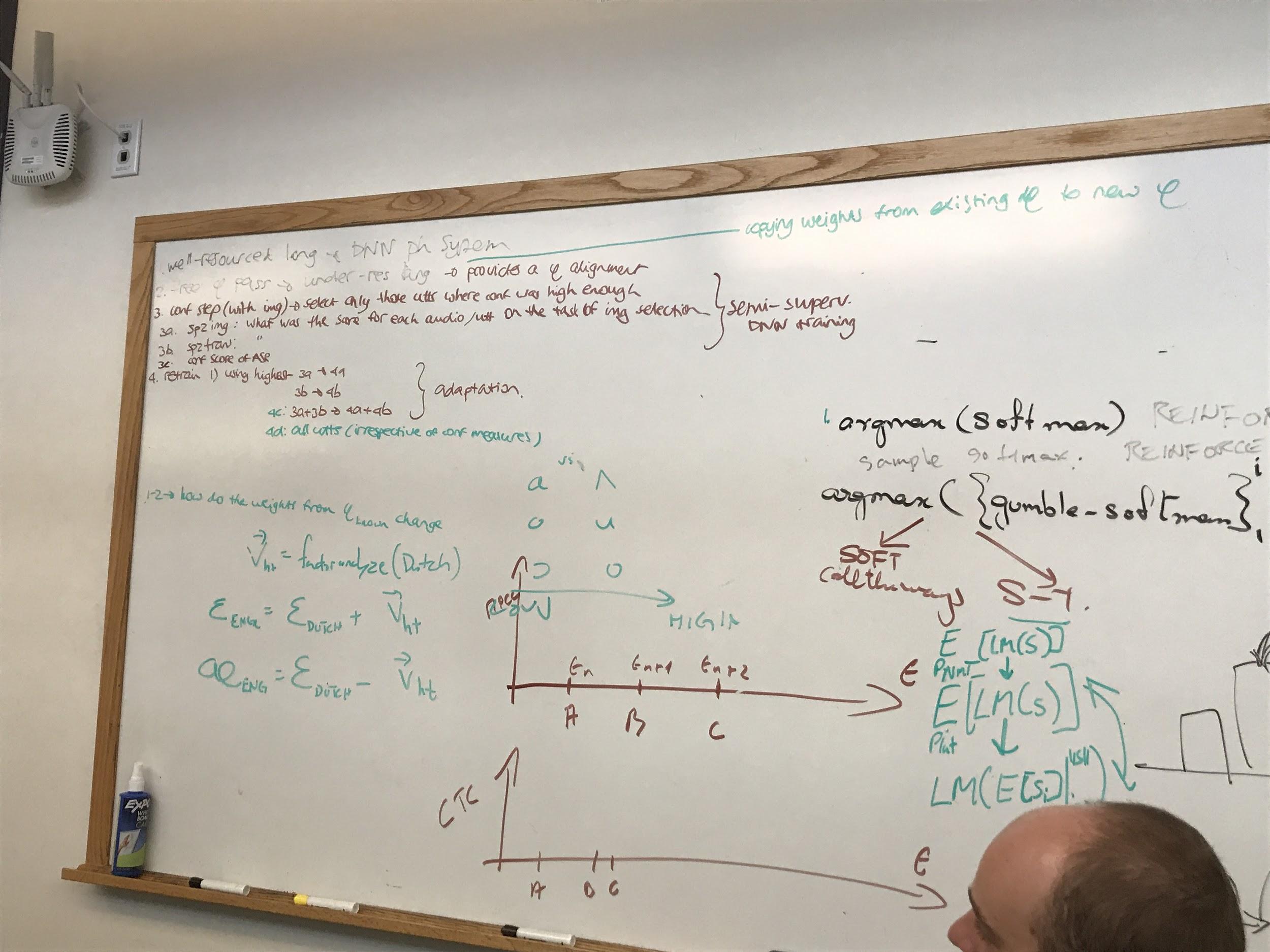
**Repeat process of RQ 1 and 2 with**

Well-resourced language: English

Under-resourced language: Mboshi

Evaluation: phone-sequence-to-translated-text retrieval task (we should assume there is no gold standard phone alignment).

**White board discussion on Monday 7/17/17**



**White board discussion on Tuesday 7/18/17**

