

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
Επεξεργασία Φωνής και Φυσικής Γλώσσας Χειμερινό εξάμηνο 2020 - 2021
Προπαρασκευή 2ου Εργαστηρίου: Αναγνώριση φωνής με το KALDi TOOLKiT

ΦΟΙΤΗΤΕΣ: ΘΕΟΔΩΡΟΠΟΥΛΟΣ ΒΑΣΙΛΕΙΟΣ
ΤΣΑΚΑΝΙΚΑ ΧΡΙΣΤΙΝΑ

AM: 03117092

AM: 03117012

Στην παρούσα αναφορά θα περιγραφεί η τεχνική διαδικασία που ακολουθήσαμε, προκειμένου να εγκαταστήσουμε το *KALDI*, ενώ θα σχολιαστούν και τα υπόλοιπα βήματα της προπαρασκευής.

1. Εγκατάσταση Kaldi

Ξεκινούμε εγκαθιστώντας το *KALDI* με την εντολή:

git

clone <https://github.com/kaldi-asr/kaldi.git>

Εγκαθιστούμε επιπλέον και τα απαραίτητα πακέτα:

```
sudo apt install -y zip python2.7 gcc g++ gfortran zlib1g-dev make automake autoconf sox libtool  
subversion gawk moreutils
```

Με την εντολή *cd kaldi* μεταβαίνουμε στον φάκελο (*directory*) *kaldi*, και με την εντολή *ls* ελέγχουμε το περιεχόμενό του. Εντοπίζουμε το αρχείο *INSTALL* και με την εντολή *cat INSTALL* το ανοίγουμε και το διαβάζουμε. Συγκεκριμένα, το αρχείο εμπεριέχει το ακόλουθο μήνυμα:

*This is the official Kaldi INSTALL. Look also at INSTALL.md for the git mirror installation.
[for native Windows [install](#), see windows/INSTALL]*

(1)

go to tools/ and follow INSTALL instructions there.

(2)

go to src/ and follow INSTALL instructions there.

Σύμφωνα λοιπόν με την υπόδειξη του μηνύματος, ακολουθούμε το βήμα 1 για την εγκατάσταση και μεταφερόμαστε στον φάκελο *tools* με την εντολή *cd tools*.

Στο directory αυτό περιέχεται το αρχείο *INSTALL* το οποίο ανοίγουμε και διαβάζουμε με την εντολή *cat INSTALL*. Το περιεχόμενο του αρχείου συνοψίζεται στις ακόλουθες υποδείξεις:

Ελέγχουμε εάν έχουν εγκατασταθεί όλα τα προαπαιτούμενα αρχεία του *Kaldi*, ευρισκόμενοι στο φάκελο *tools*, με την εντολή: *extras/check_dependencies.sh*.

Εάν υπάρχουν προαπαιτούμενα που δεν έχουν εγκατασταθεί επιτυχώς, εμφανίζεται μήνυμα στην οθόνη, επισημαίνοντάς μας την εντολή εγκατάστασής τους. Όταν πλέον έχουμε εγκαταστήσει κάθε αναγκαίο αρχείο του *Kaldi* η έξοδος που εμφανίζεται στις οθόνες μας με την εντολή *extras/check_dependencies.sh* είναι η ακόλουθη: *all OK*.

Εφόσον, ο προκαθορισμένος *python interpreter* για εμάς ήταν ο 2.7, δε δημιουργήθηκε κάποιος νέος φάκελος *python*. Κατά συνέπεια, δε χρειάστηκε να διαγράψουμε το περιεχόμενο του φακέλου αυτού, ούτε να δημιουργήσουμε εντός αυτού ένα κενό αρχείο *.use_default_python*.

Έπειτα, πρέπει να εκτελέσουμε την εντολή *make*. Προκειμένου να επισπεύσουμε τη διαδικασία αυτή, εάν το σύστημά μας διαθέτει πολλαπλές κεντρικές μονάδες επεξεργασίας (*CPUs*) το build up μπορεί να γίνει παράλληλα χρησιμοποιώντας κάθε έναν από τους επεξεργαστές. Η εντολή *nproc* εμφανίζει το πλήθος των επεξεργαστών, πχ 4, και βασιζόμενοι σε αυτό, πληκτρολογούμε την εντολή: *make -j 4*.

Παρευρισκόμενοι στον φάκελο *tools*, εγκαθιστούμε το πακέτο *IRSTLM* χάρη στην εντολή:
extras/install_irstlm.sh

Στη συνέχεια, εγκαθιστούμε και το πακέτο *OpenBLAS* με την εντολή *extras/install_openblas.sh*.

Συνεπώς, έχοντας πλέον ολοκληρώσει το πρώτο βήμα του αρχείου *INSTALL* συνεχίζουμε με το δεύτερο βήμα. Σύμφωνα με αυτό μεταβαίνουμε, λοιπόν, στον φάκελο *src* από τον φάκελο *tools* με την εντολή: *cd ../src/*.

Στον φάκελο *src* υπάρχει ένα ακόμα αρχείο *INSTALL* το οποίο διαβάζουμε με την εντολή *cat INSTALL*. Τα βήματα εγκατάστασης που μας υποδεικνύει το αρχείο είναι τα ακόλουθα:

Αρχικά πρέπει να πληκτρολογήσουμε την εντολή *./configure* στην γραμμή εντολών. Για τις ανάγκες της εργαστηριακής άσκησης τροποποιούμε την παραπάνω εντολή στην ακόλουθη μορφή: *./configure --shared --openblas-root=../tools/OpenBLAS/install*

Εν συνεχεία, πρέπει να τρέξουμε τις εντολές *make depend* και *make*. Αντίστοιχα με πριν, προκειμένου να ολοκληρωθεί η ενέργεια ταχύτερα, ανάλογα με τους επεξεργαστές που διαθέτει το σύστημά μας, πχ 4, εν τέλει τρέχουμε τις εντολές *make depend -j 4* και *make -j 4*.

Σημειώνοντας τα παραπάνω βήματα, ολοκληρώνεται επιτυχώς η εγκατάσταση της εργαλειοθήκης *KALDI*.

2. Εξοικείωση με το εργαλείο Kaldi

Στο *directory* *egs* υπάρχουν πολλά παραδείγματα που αναπαριστούν πώς μπορούμε να χρησιμοποιήσουμε το *kaldi*, ενώ κάθε *subdirectory* του, ανταποκρίνεται σε ένα λεξικό (*corpus*) για το οποίο διαθέτουμε κάποια *scripts* ως παραδείγματα.

Προκειμένου, λοιπόν, να εξοικειωθούμε με το εργαλείο *Kaldi* μεταβαίνουμε από το φάκελο *egs* στο φάκελο *yesno* και από εκεί στο φάκελο *s5*. Ο φάκελος *s5* περιλαμβάνει 6 ακόμα φακέλους και 16 αρχεία με γενικές πληροφορίες που αφορούν τη γλώσσα (*input/ dir*) και τις οδηγίες προκειμένου να προετοιμάσουμε τα δεδομένα (*local/ dir preparing the data and scoring it*). Περιέχουν ακόμα, πληροφορίες για το είδος του μοντέλου που επιθυμούμε να εκπαιδεύσουμε και να ελέγξουμε (*conf/ dir train and test*). Τέλος, βοηθητικά *scripts* που πραγματεύονται την εκπαίδευση *monophones* και λήψη *MFCCs* από *WAV files* εντοπίζονται στους φακέλους *steps/* και *utils/*.

Ευρισκόμενοι, λοιπόν, στον φάκελο *s5* πληκτρολογούμε στην γραμμή εντολών την εντολή *./run.sh* η οποία παράγει 18 ακόμα *directories* και κάποια επιπλέον αρχεία. Επιπλέον, παράγονται και τα *directories data* και *exp* τα οποία περιέχουν και οργανώνουν τα αρχεία που περιγράφουν τη γλώσσα

(πχ. λεξικό, λίστα τηλεφώνου) καθώς και δεδομένα (πχ αρχεία *WAV* με τα αντίστοιχα *id* τους) για την εκπαίδευση και τον έλεγχο του μοντέλου. Τέλος, ο φάκελος *mfcc* συγκεντρώνει όλα τα *.ark* και *.spc* αρχεία, ενώ συγκαταλέγει πληροφορίες για τα *MFCC* χαρακτηριστικά και την *cepstral* μέση τιμή και διακύμανση κάθε ομιλητή.

3. Εγκατάσταση προαπαιτούμενων αρχείων για την εργαστηριακή άσκηση

Εγκαθιστούμε το αρχείο *slp_lab2_data* στο *directory /kaldi/egs/usc/data* με την ακόλουθη εντολή:

```
wget https://drive.google.com/file/d/1M5p17VySiWANsKfuZyaXs9i2vPer0mTd/view?usp=sharing
```

4. Κατασκευή αρχικού σκελετού

Αρχικά δημιουργούμε τους φακέλους *train*, *dev*, *test* με μονοπάτια: *egs/usc/data/train*, *egs/usc/data/dev*, *egs/usc/data/test* αντίστοιχα. Δηλαδή οι τρεις αυτοί φάκελοι βρίσκονται στον ίδιο φάκελο με το *slp_lab2_data*.

Έπειτα σε κάθε έναν από αυτούς τους υποφακέλους δημιουργούμε τα αρχεία

uttdids: Περιέχει στην κάθε του γραμμή ένα μοναδικό συμβολικό όνομα για κάθε πρόταση του συγκεκριμένου συνόλου δεδομένων. Το περιεχόμενο του αρχείου αυτού ταυτίζεται με το περιεχόμενο του αρχείου εντός του φακέλου *filesets*, για τον αντίστοιχο φάκελο (είτε *train*, είτε *test*, είτε *dev*).

utt2spk: Περιέχει σε κάθε γραμμή τον ομιλητή που αντιστοιχεί σε κάθε πρόταση και είναι της μορφής: *utterance_id_1* <κενό> *speaker_id*
utterance_id_2 <κενό> *speaker_id* κ.ο.κ.

Προκειμένου να παράξουμε το αρχείο αυτό, γράφουμε κώδικα σε python. Συγκεκριμένα, το πρόγραμμά μας διαβάζει κάθε ένα εκ των *utterances.txt* αρχείων εντός του *filesets* απομονώνει την κάθε γραμμή του αρχείου και από αυτή τον ομιλητή που της αντιστοιχεί (*f1*, *f5*, *m1*, *m3*), τον οποίο και τυπώνει μετά το *utterance_id*.

wav.scp:

περιέχει τη θέση του αρχείου ήχου που αντιστοιχεί σε κάθε πρόταση και είναι της μορφής: *utterance_id_1* <κενό> */path/to/wav1*
utterance_id_2 <κενό> */path/to/wav2*

Για τα συγκεκριμένα αρχεία διαβάζουμε κάθε αρχείο του *fileset* και τυπώνουμε το *utterance_id* σε κάθε περίπτωση με το μονοπάτι του *wav* αρχείου που είναι της μορφής *slp_lab2_data/wav/f1/usc_timit_ema_f1_258.wav*

text:

Περιέχει το κείμενο που αντιστοιχεί στην κάθε πρόταση και είναι της μορφής:
utterance_id_1 <κενό> <utterance 1 text>

utterance_id_2 <κενό> <utterance 2 text>

Στο πρόγραμμά μας, διαβάζοντας κάθε ένα από τα αρχεία στο `filesets`, ανάλογα με τον αριθμό που έχει η κάθε πρόταση στο τέλος της, τυπώνουμε την αντίστοιχη πρόταση από το αρχείο `transcription.txt`, σύμφωνα με τις παραπάνω μορφές. Για παράδειγμα, αν η πρόταση στα `filesets` είναι η `usctimit_ema_f1_001`, τότε ο ζητούμενος αριθμός είναι το 001. Συνεπώς τυπώνουμε την πρώτη πρόταση του αρχείου `transcription.txt`.

Τέλος, αντικαθιστούμε τις λέξεις των προτάσεων με τα αντίστοιχα φωνήματα, αφού έχουμε αντιστοιχίσει στον κώδικά μας κάθε λέξη του λεξικού `lexicon.txt` με το φώνημά της.

Οι κώδικες για την παραγωγή κάθε ενός από τα παραπάνω αρχεία παρατίθενται στον φάκελο *scripts* εντός του υποβιβλημένου μας αρχείου *zip*.