ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Επεξεργασία Φωνής και Φυσικής Γλώσσας

Χειμερινό εξάμηνο 2020 - 2021

2ο Εργαστήριο: Αναγνώριση φωνής με το KALDi TOOLKiT

Θεοδωρόπουλος Βασίλειος AM: 03117092
Τσακανίκα Χριστίνα AM: 03317012

Βήματα κυρίως μέρους

4.1 Προετοιμασία διαδικασίας αναγνώρισης φωνής για τη USC-TIMIT

Στο συγκεκριμένο βήμα της εργαστηριακής άσκησης διαμορφώνουμε το *directory structure* που θα αποτελέσει τη βάση του συστήματος αναγνώρισης φωνής. Έχοντας ήδη δημιουργήσει τον φάκελο εργασίας *usc,* στην προπαρασκευή (prelab.sh), μεταφέρουμε σε αυτόν τα αρχεία path.sh & cmd.sh. Στο αρχείο *path* καλούμαστε να τροποποιήσουμε την τιμή της μεταβλητής "KALDI_ROOT" από "KALDI_ROOT=`pwd`/../../.." σε "KALDI_ROOT=`pwd`/../.." όπου οι χαρακτήρες "/../.." συμβολίζουν ότι ο κεντρικός φάκελος εγκατάστασης του *kaldi* βρίσκεται δύο φακέλους πιο πάνω του δικού μας φακέλου *usc*. Η αντιγραφή των αρχείων στο φάκελο εργασίας έγινε βάσει της εντολής "cp path/to/file name/of/new/file", η δημιουργία των *soft links* βάσει της εντολής "ln -s path/to/file" και τέλος για την δημιουργία νέων *directories* χρησιμοποιούμε την εντολή "mkdir".

4.2 Προετοιμασία γλωσσικού μοντέλου

Τα βασικά αρχεία για τη δημιουργία του γλωσσικού μοντέλου δημιουργούνται σύμφωνα με το script ex4.2.1.py.
Με την εντολή *build-lm* εκπαιδεύουμε ένα n-gram γλωσσικό μοντέλο, μετράμε τις πιθανότητες εμφάνισης n-grams και τέλος το σώζουμε με την κατάληξη .ilm.gz. Η εντολή *compile-lm* έχει ως είσοδο το αρχείο .ilm.gz και το αποθηκεύει στην μορφή *arpa.gz*.
Οι δύο αυτές εντολές συνθέτουν το L.fst, το λεξικό δηλαδή του συστήματος, όπου έχει γίνει η αντιστοίχιση εκφωνήσεων σε fst's states.
Το *kaldi* επιθυμεί τα αρχεία wav.scp, text και utt2sp σε ταξινομημένη μορφή για το λόγο αυτό εκτελούμε την εντολή sort αρχικό/αρχείο -o τελικό/αρχείο
Το script *utt2spk_to_spk2utt.pl* δημιουργεί σε κάθε έναν από τους φακέλους dev, train, test το αρχείο spk2utt.
Το τελικό script που τρέχουμε είναι το timit_format_data.sh, προκειμένου να μορφοποιήσουμε τα δεδομένα. Το script αυτό με τη σειρά του εκτελεί το validate_data_dir.sh, το οποίο ελέγχει τα directories που έχουμε δημιουργήσει και αποφαίνεται για τη συμβατότητά τους με το *kaldi*. Έπειτα, δημιουργείται ο γράφος G. Συγκεκριμένα, το γλωσσικό μοντέλο που έχει εκπαιδευτεί, γίνεται compile σε open fst format ("κοιτάει" ένα στοχαστικό μοντέλο) και προκύπτουν τα αρχεία L.fst & G.fst που αποτελούν το δεύτερο τμήμα του γράφου HCLG.

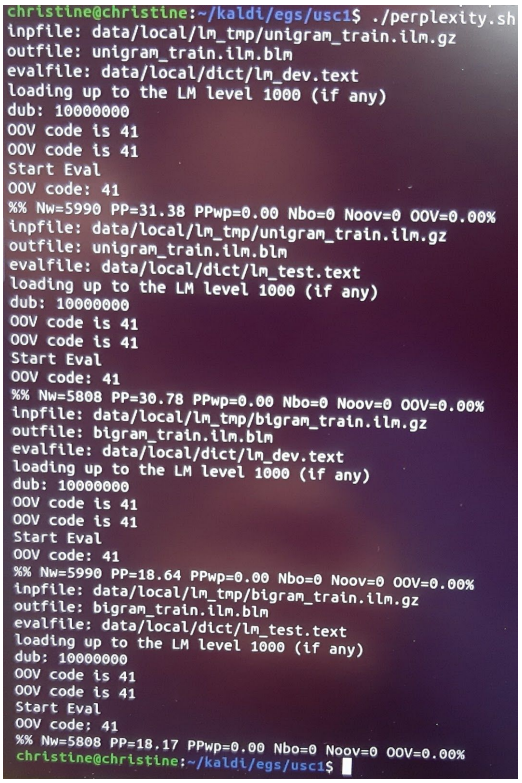
Ερώτημα 1

Perplexity of unigram model in validation set: 31.38 OOV: 0.00%

Perplexity of unigram model in test set: 30.78 OOV: 0.00%

Perplexity of bigram model in validation set: 18.64 OOV: 0.00%

Perplexity of bigram model in test set: 18.17 OOV: 0.00%



Γλωσσική Πολυπλοκότητα:

Με το γλωσσικό μοντέλο κάθε προβλήματος, συσχετίζεται ένα μέτρο που εκφράζει πόσο σύνθετο είναι το μοντέλο και είναι γνωστό ως πολυπλοκότητα γλώσσας. Η πολυπλοκότητα της γλώσσας είναι ο γεωμετρικός μέσος του παράγοντα διακλάδωσης λέξεων. Η πολυπλοκότητα της γλώσσας όπως ενσωματώνεται σε ένα γλωσσικό μοντέλο $P_L(w)$ για μια ακολουθία Μ λέξεων υπολογίζεται από την εντροπία ως:

$$H(w) = -\frac{1}{M}\log_2 P(w)$$

4.3 Εξαγωγή ακουστικών χαρακτηριστικών

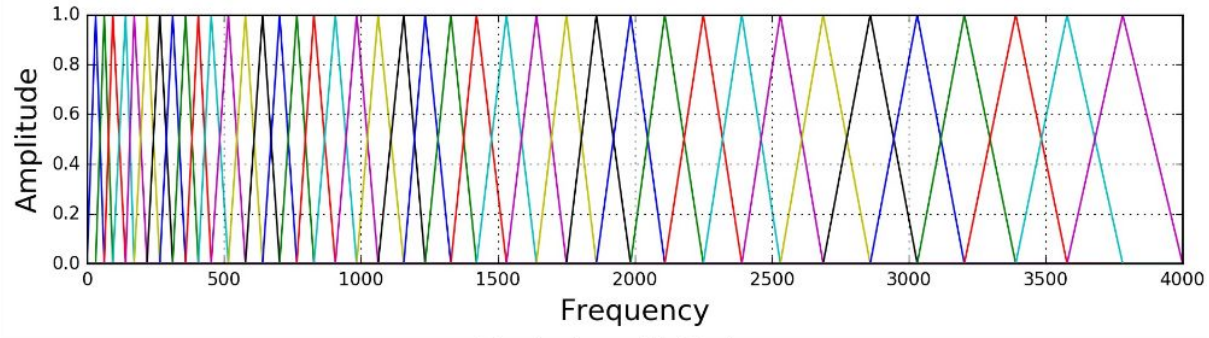
Η εξαγωγή των χαρακτηριστικών MFCC's επιτυγχάνεται μέσω των scripts *make_mgcc.sh* & *compute_cmvn_stats.sh*.
Τα MFCC's χαρακτηριστικά είναι αυτά που προκύπτουν από τα σήματα ήχου. Έχοντας ως είσοδο ένα σήμα ήχου, το χωρίζω σε frames και εφαρμόζω ένα pre-emphasis φίλτρο.

Εν συνεχεία, χωρίζω το σήμα σε επικαλυπτόμενα παράθυρα, εφαρμόζω Μετασχηματισμό Fourier του σήματος, έπειτα το σήμα διέρχεται από μία "τράπεζα" Mel-φίλτρων. Οι Mel συχνότητες και οι μετατροπή τους σε Hertz προσδιορίζονται από την σχέση:

$$m = 2595 \log_{10}(1 + \frac{f}{700})$$

$$f = 700(10^{m/2595} - 1)$$

Κάθε φίλτρο της τράπεζας είναι τριγωνικό λαμβάνοντας πλάτος 1 στην κεντρική συχνότητα, ενώ μειώνεται γραμμικά μέχρι το μηδέν έως ότου να φτάσει τις κεντρικές συχνότητες των γειτονικών φίλτρων.



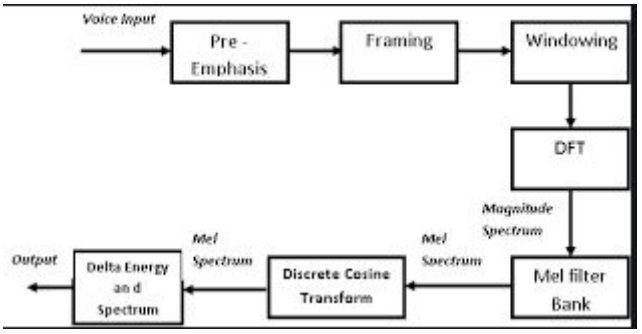
Filter bank on a Mel-Scale

Παρατηρούμε ότι το MEL δίνει έμφαση στις χαμηλές συχνότητες, δίχως κάποια μαθηματική απόδειξη, παρά μόνο βασιζόμενο σε ψυχοακουστικές μελέτες, οι οποίες απέδειξαν ότι το ανθρώπινο αυτί αντιλαμβάνεται ευκολότερα τις χαμηλές συχνότητες. Πράγματι, το παραπάνω διάγραμμα παρουσιάζει πυκνότερη μοντελοποίηση χαμηλών συχνοτήτων, ενώ όσο αυξάνονται οι συχνότητες το διάγραμμα αραιώνει.

Εν συνεχεία, με την εφαρμογή λογαρίθμου και DCT λαμβάνουμε το cepstrum του σήματος. Το cepstrum είναι ένα πεδίο παρόμοιο του χρόνου, το οποίο αποτυπώνει την περιβάλλουσα του σήματος και διευκολύνει την την επεξεργασία των σημάτων ήχων.

Καταλήγουμε εν τέλει με 13 χαρακτηριστικά (features) στα οποία παίρνοντας τις πρώτες παραγώγους ή ακόμα και τις δεύτερες παραγώγους συνθέτουμε 39 features για ένα frame.

Τα παραπάνω περιγράφονται διαγραμματικά ως ακολούθως:



Ερώτημα 2

Cepstral Mean and Variance Normalization είναι μία υπολογιστική μέθοδος κανονικοποίησης για σήματα ήχου. Συγκεκριμένα, υπολογίζουμε το cepstrum, αφαιρούμε τον μέσο όρο όλων των frames. Μαθηματικά τα παραπάνω περιγράφονται ως εξής:

Σήμα εισόδου $x[n]$ και κρουστική απόκριση συστήματος $h[n]$. Έξοδος y[n]:

$$y[n] = x[n] \star h[n]$$

Μετασχηματισμός Fourier:

$$Y[f] = X[f] \cdot H[f]$$

Λογαριθμούμε (Cepstrum):

$$Y[q] = \log Y[f] = \log(X[f] \cdot H[f]) = X[q] + H[q]$$

Για κάθε i-στο frame ισχύει:

$$Y_i[q] = H[q] + X_i[q]$$

Ενώ λαμβάνοντας το μέσο όρο κάθε frame:

$$\frac{1}{N}\sum_i Y_i[q] = H[q] + \frac{1}{N}\sum_i X_i[q]$$

ορίζουμε τη διαφορά

$$\begin{aligned} R_i[q] &= Y_i[q] - \frac{1}{N}\sum_j Y_j[q] \\ &= H[q] + X_i[q] - \left(H[q] + \frac{1}{N}\sum_j X_j[q]\right) \\ &= X_i[q] - \frac{1}{N}\sum_j X_j[q] \end{aligned}$$

Αυτού του είδους η κανονικοποίηση δεν είναι υποχρεωτική, ιδίως όταν έχουν συστήματα ενός μόνο εκφωνητή σε ένα μόνο περιβάλλον. Συγκεκριμένα, εάν το σήμα έχει θόρυβο, τα αποτελέσματα που θα λάβουμε μετά από αυτήν την κανονικοποίηση θα είναι υποδεέστερα, από αυτά δίχως την κανονικοποίηση, όπως επισημαίνεται με τον κόκκινο όρο.

$$y[n] = x[n] \star h[n] + w[n]$$

$$Y[f] = X[f] \cdot H[f] + W[f]$$

$$\log Y[f] = \log\left[X[f]\left(H[f] + \frac{W[f]}{X[f]}\right)\right] = \log X[f] + \log\left(H[f] + \frac{W[f]}{X[f]}\right)$$

Σε συνθήκες SNR ο σημειωμένος με κόκκινο όρος μπορεί να ξεπεράσει την εκτίμηση.

Από την άλλη, η εφαρμογή του CMS αυξάνει την απόδοση του συστήματος, συγκεκριμένα αυξάνει ο ρυθμός αναγνώρισης.

Ερώτημα 3

Ανοίγουμε το αρχείο utt2num_frames του φακέλου train που δημιουργήθηκε. Αντιγράφουμε το περιεχόμενό του για τις πέντε πρώτες προτάσεις

usctimit_ema_f1_001 237
usctimit_ema_f1_002 377
usctimit_ema_f1_003 317
usctimit_ema_f1_005 399
usctimit_ema_f1_006 338

Άρα, τα πλαίσια για την κάθε μία πρόταση κατά σειρά είναι 237, 377, 317, 399, 338.
Όπως αναλύθηκε και πρωτύτερα, σε κάθε frame αντιστοιχούν 39 χαρακτηριστικά, που μπορούν να παρομοιαστούν με την διάστασή του.

4.4 Εκπαίδευση ακουστικών μοντέλων και αποκωδικοποίηση προτάσεων

Το μονοφωνικό GMM-HMM γίνεται train με το script steps/train_mono.sh. Εφόσον, το μονοφωνικό αυτό μοντέλο δεν περιέχει καποιο είδος align η πρόβλεψή του δε θα είναι βάσιμη. Έπειτα με το script utils/mkgraph.sh δημιουργείται το H και ένας identity μετασχηματισμός για το C, ο οποίος δεν υλοποιεί εξαρτήσεις μεταξύ φωνημάτων. Οπότε έχοντας ήδη ένα L, G για unigram μοντελοποίηση και ένα L,G για bigram μοντελοποίηση έπειτα από την σύνθεση HCLG δημιουργείται ένα μονοφωνικό μοντέλο unigram και έπειτα ένα μονοφωνικό μοντέλο για bigram. Τέλος, τρέχω το decode script για να προκύψουν τα ανάλογα phoneme errors. Συγκεκριμένα:

Για το μονοφωνικό μοντέλο οι καλύτερες επιδόσεις είναι:

unigram_validation: %WER 54.37 [3356 / 6173, 76 ins, 1852 del, 1428 sub]

unigram_test: %WER 54.37 [3356 / 6173, 76 ins, 1852 del, 1428 sub]

bigram_validation: %WER 51.26 [3164 / 6173, 83 ins, 1577 del, 1504 sub]

bigram_test: : %WER 48.63 [2914 / 5992, 61 ins, 1439 del, 1414 sub]

Αυτό το μονοφωνικό μοντέλο δεν αποτελεί το τελικό μοντέλο αναγνώρισης φωνής, αλλά χρησιμοποιείται για να κάνω καλό alignment των φωνημάτων με τα με τα αρχεία ήχου, ούτως ώστε έπειτα να εκπαιδεύσω το τριφωνικό μοντέλο. Δεδομένων, άρα, των αρχείων ήχου που μπορεί να περιέχουν θόρυβο, περιόδους σιωπής και ένα transcription, βασικό βήμα αποτελεί το align. Έτσι καθορίζονται οι χρονικές στιγμες που αντιστοιχεί ένα φώνημα. Χρησιμοποιώ αρχικά το script steps/align_si.sh . Έπειτα, κάνω train το τριφωνικό μοντέλο πάνω σε αυτά τα alignments, μέσω της εντολής steps/train_deltas.sh. Η εκπαίδευση του τριφωνικού μοντέλου γίνεται αρκετές φορές έως ότου να πάψω να παρατηρώ σημαντική βελτίωση. Τέλος, κάνω και πάλι decoding, όπου χάρη στον

viterbi αλγόριθμο, γίνεται υπολογισμός του πιθανότερου φωνήματος για το τρέχον utterance και εν τέλει δημιουργείται ο γράφος HCLG. Συνεπώς, προκύπτει ένα τριφωνικό μοντέλο για unigram και ένα τριφωνικό μοντέλο για bigram υλοποίηση. Τέλος, υπολογίζεται ξανά το phoneme error rate. Το τριφωνικό, αυτό, μοντέλο αποτελεί ένα αποδοτικό μοντέλο αναγνώρισης φωνής.

Για το τριφωνικό μοντέλο οι καλύτερες επιδόσεις είναι:

unigram validation: %WER 41.62 [2569 / 6173, 205 ins, 885 del, 1479 sub]

unigram_test: %WER 39.20 [2349 / 5992, 144 ins, 893 del, 1312 sub]

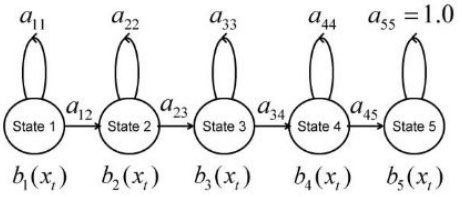
bigram_validation: %WER 39.75 [2454 / 6173, 214 ins, 855 del, 1385 sub]

bigram_test: %WER 36.60 [2193 / 5992, 160 ins, 801 del, 1232 sub]

Παρατηρούμε ότι οι επιδόσεις του τριφωνικού μοντέλου είναι αισθητά βελτιωμένες του μονοφωνικού, όπως ακριβώς αναμέναμε. Η καλύτερη, όλων των επιδόσεων, σημειώθηκε στο set dev για το bigram τριφωνικό μοντέλο (%WER 36.60). Οι υπερπαράμετροι του script score.sh φαίνεται να είναι hyp_filtering_cmd και wer_hyp_filter.

Ερώτημα 4

Η πιο συχνή μέθοδος κατασκευής ακουστικών μοντέλων, τόσο για φωνήματα όσο και λέξεις, είναι το στατιστικό κατασκεύασμα γνωστό ως *κρυφό μοντέλο Markov* (*HMM*). Κάθε κατάσταση του HMM χαρακτηρίζεται από μία μίξη *Gaussian* πυκνοτήτων πιθανότητας (GMM), η οποία αναπαριστά τη στατιστική συμπεριφορά των διανυσμάτων χαρακτηριστικών Xt, στις καταστάσεις του μοντέλου. Επιπλέον, το HMM χαρακτηρίζεται επίσης από ένα σύνολο μεταβάσεων μεταξύ καταστάσεων, $A=\{a_{ij}, 1\leq i,j\leq Q\}$, για ένα μοντέλο Q καταστάσεων, οι οποίες περιγράφουν την πιθανότητα να γίνει μία μετάβαση από την κατάσταση i στην κατάσταση j σε κάθε πλαίσιο, καθορίζοντας έτσι τη χρονική ακολουθία των διανυσμάτων χαρακτηριστικών κατά την διάρκεια της λέξης. Συνήθως, οι μεταβάσεις από μία κατάσταση στον εαυτό της, a_{ij}, είναι μεγάλες (πιθανότητα κοντά στο 1), ενώ οι καταστάσεις άλματος $a_{12},a_{23},a_{34},a_{45}$ είναι μικρές (πιθανότητα κοντά στο μηδέν). Το πλήρες *HMM* μίας λέξης Q καταστάσεων γράφεται στη γενική περίπτωση ως $\lambda(A,B,\pi)$, με μητρώο μετάβασης καταστάσεων $A=\{a_{ij}, 1\leq i, j\leq Q\}$, πυκνότητα πιθανότητα παρατήρησης εκπομπών ανά κατάσταση $B=\{b_j(x_t), 1\leq j\leq Q\}$ και κατανομή αρχικής κατάστασης, $\pi=\{\pi_i, 1\leq i\leq Q\}$, όπου το π_1 τίθεται ίσο με ένα και όλα τα υπόλοιπα τίθενται ίσα με μηδέν, για τα μοντέλα είδους αριστερά προς τα δεξιά, όπως αυτό του παρακάτω σχήματος.



Το παραπάνω μοντέλο αποτελεί *HMM* με πέντε καταστάσεις και μηδενικές πιθανότητες υπερπήδησης καταστάσεων, δηλαδή $a_{ij=0, j\geq i+2}$. Η συνάρτηση πυκνότητας για την κατάσταση i δηλώνεται ως $b_i(X_t)$, $1 \leq i \leq 5$.

Προκειμένου να γίνει η εκπαίδευση του *HMM* κάθε λέξης χρησιμοποιείται ένα επισημειωμένο σύνολο προτάσεων εκπαίδευσης, με σκοπό να ακολουθηθεί μία αποτελεσματική διαδικασία εκπαίδευσης βάσει του αλγορίθμου *Viterbi.* Ο αλγόριθμος αυτός έχει ως στόχο την εύρεση διαδρομής μέγιστης πιθανότητας που συνδέει κάθε διάνυσμα χαρακτηριστικών της ομιλίας $X=\{x_1,x_2,...,x_T\}$, με μία μοναδική κατάσταση μοντέλου. Ορίζοντας την ποσότητα $\delta_t(j)$, ως την πιθανότητα να βρεθεί η βέλτιστη διαδρομή στην κατάσταση *i*, στο χρονικό πλαίσιο *t*, αφού έχουν παρέλθει t διανύσματα της πρότασης και δοθέντος του μοντέλου λ

$$\delta_t(j) = \max P[q_1, q_2, q_3, \dots q_{t-1}, q_t = i, X1, X2, X3,... Xt| \lambda]$$

Ο αλγόριθμος *Viterbi* υπολογίζει τη διαδρομή υψηλότερης πιθανότητας που λαμβάνει υπόψη όλα τα Τ διανύσματα χαρακτηριστικών της πρότασης, μέσω αναδρομής.

Όπως αναλύθηκε και στο εργαστήριο, κατά την εκπαίδευση του μονοφωνικού μοντέλου ακολουθείται η ακόλουθη διαδικασία:
Αρχικά, εκπαιδεύουμε ένα HMM-GMM μονοφωνικό μοντέλο, δίχως alignment. Στην συνέχεια, πάνω σε αυτό το μοντέλο κάνω alignment προκειμένου να εξαχθούν τα ζητούμενα alignments. Στο επόμενο βήμα, εκπαιδεύω ένα μονοφωνικό μοντέλο πάνω σε αυτά τα alignments 20-30 φορές, όσες τρέχει στην πραγματικότητα το script train_mono. Κατά συνέπεια, το τελικό μονοφωνικό που προκύπτει αποτελεί ένα αποδοτικό μοντέλο, πάνω στο οποίο μπορώ να εκπαιδεύσω και το τριφωνικό μου μοντέλο.

Ερώτημα 5

Το κεντρικό πρόβλημα της Αυτόματης Αναγνώρισης Λόγου (Automatic Speech Recognition), αντιμετωπίζεται ως ένα στατιστικό πρόβλημα απόφασης. Συγκεκριμένα, διατυπώνεται ως μία διεργασία απόφασης με βάση την μέγιστη εκ των υστέρων πιθανότητα, όπου αναζητούμε την ακολουθία λέξεων W που μεγιστοποιεί τη εκ των υστέρων πιθανότητα P(W|X), της ακολουθίας με δεδομένη την ακολουθία των διανυσμάτων χαρακτηριστικών, X, δηλαδή:

$$\hat{W} = \arg\max_W P(W|X)$$

Εφαρμόζοντας τον κανόνα του Bayes στην παραπάνω σχέση και και αγνοώντας τον όρο του παρονομαστή, P(X), διότι είναι ανεξάρτητος από την ακολουθία λέξεων W, ως προς την οποία γίνεται η βελτιστοποίηση, λαμβάνουμε διαδοχικά τις σχέσεις:

$$\hat{W} = \arg\max_W \frac{P(X|W)P(W)}{P(X)}.$$

$$\hat{W} = \underbrace{\arg\max_W}_{\text{Step 3}} \underbrace{P_A(X|W)}_{\text{Step 1}} \underbrace{P_L(W)}_{\text{Step 2}}$$

Εκτενέστερα και τεκμηριωμένα βήματα για τον υπολογισμό της πιθανότητας αυτής παρατίθενται στο αρχείο "Fundamentals_of_SLP" που βρίσκεται στα παραδοτέα μας.

Ερώτημα 6

Ο γράφος HCLG στο *kaldi* προκύπτει από τις συνθέσεις HCLG = H ο C ο L ο G. Συγκεκριμένα,

Ο γράφος G είναι ένας αποδοχέας (τα σύμβολα εισόδου του ταυτίζονται με τα σύμβολα εξόδου)

Το L είναι ένα λεξικό, όπου τα σύμβολα εισόδου του είναι φωνήματα και τα σύμβολα εξόδου του είναι λέξεις.

Ο γράφος C στην περίπτωσή μας είναι identity, εφόσον δεν υλοποιεί εξαρτήσεις μεταξύ των φωνημάτων

Το H περιλαμβάνει τους ορισμούς του HMM.