ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

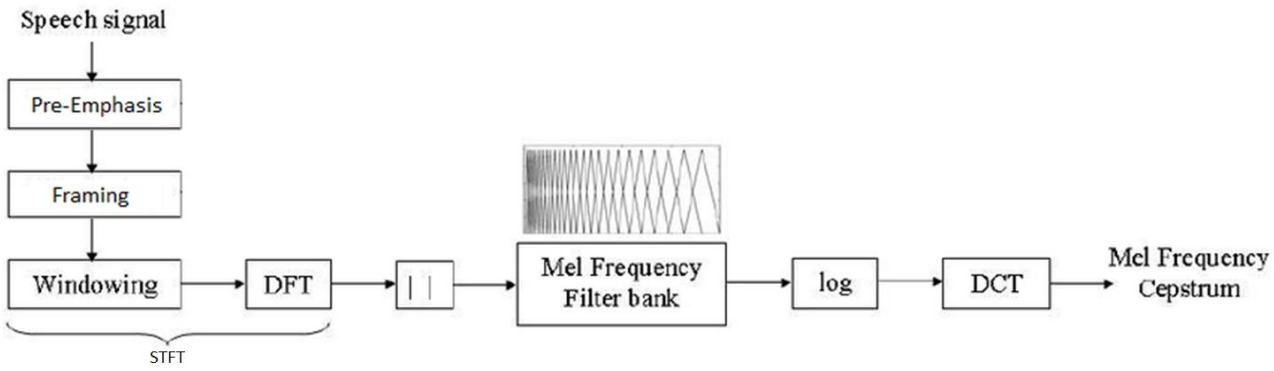
Επεξεργασία Φωνής και Φυσικής Γλώσσας
Χειμερινό Εξάμηνο 2020-2021
2η Εργαστηριακή Άσκηση: Αναγνώριση φωνής με το KALDI TOOLKIT

Φοιτητές: Λιακόπουλος Δημήτριος (03117109)
Μπεκρής Δημήτριος (03117116)

ΘΕΩΡΗΤΙΚΗ ΑΝΑΛΥΣΗ

Πριν γίνει η αναλυτική περιγραφή των βημάτων που ακολουθήθηκαν κατά την εργαστηριακή άσκηση, θεωρείται δόκιμο να αναλυθούν ορισμένοι βασικοί ορισμοί που εμφανίζονται. Αυτοί οι ορισμοί είναι οι ακόλουθοι:

- ***MFCCs***: Οι MFCCs (Mel-Frequency Cepstrum Coefficients) είναι συντελεστές που εισήχθησαν από τους Davis και Mermlstein ως ένας νέος, στους μέχρι στιγμής υπάρχοντες, τύπος αναπαράστασης Cepstrum.
  Η διαφορά μεταξύ του Cepstrum και του Mel-Frequency Cepstrum εντοπίζεται στο γεγονός ότι στο δεύτερο οι συχνότητες τοποθετούνται στην mel-κλίμακα, ούτως ώστε να προσεγγίζεται η ακουστική του ανθρώπινου αυτιού, όπου είναι ευαίσθητο στην ανίχνευση μεταβολών στις χαμηλές συχνότητες αλλά όχι στις υψηλές.
  Η διαδικασία παραγωγής των mfccs είναι ίδια με αυτήν που ακολουθείται για την εξαγωγή των filter banks με την διαφορά ότι στο τέλος συμπεριλαμβάνονται μερικά επιπλέον βήματα. Συνοπτικά η διαδικασία περιλαμβάνει την προ-έμφαση (pre-emphasis) του σήματος, όπου στην συνέχεια διαχωρίζεται σε πλαίσια (frames), τα οποία περνάνε από παράθυρα (windows). Στην συνέχεια, σε κάθε πλαίσιο εκτελείται ένας μετασχηματισμός Fourier και έτσι υπολογίζονται τα filter banks. Τέλος, με ένα μετασχηματισμό DCT (Dicrete Cosine Transform) παράγονται οι mfccs, όπου κανονικοποιούνται για να είναι τα αποτελέσματα πιο συνεπή. Η περιγραφή αυτή αναπαριστάται και στο ακόλουθο σχεδιάγραμμα σχηματικά:



Αναλυτικά η παραπάνω διαδικασία είναι η ακόλουθη:

- ***Προ-έμφαση (Pre-Emphasis)***
  Αρχικά, περνάμε το σήμα, έστω x(t), από ένα pre-emphasis φίλτρο για να ενισχυθούν οι υψηλές συχνότητες. Επιπρόσθετα, το φίλτρο αυτό κάνει μια κανονικοποίηση του spectrum έτσι ώστε το πλάτος των υψηλών και των χαμηλών συχνοτήτων να μην έχουν τεράστιες αποκλίσεις, βοηθά στην προετοιμασία των δεδομένων ούτως ώστε να αποφευχθούν τα λάθη κατά τον μετασχηματισμό Fourier και, τέλος, βελτιώνει το SNR (Signal-to-Noise Ratio).
  Το pre-emphasis φίλτρο μοντελοποιείται μαθηματικά με τον ακόλουθο τύπο:
  $$y(n) = x(n) - a * x(n-1)$$, όπου α = 0.95 - 0.97

- ***Πλαισιοποίηση (Framing)***
  Σε αυτό το βήμα γίνεται η διαίρεση του σήματος σε πλαίσια. Αυτό το βήμα, είναι απαραίτητο διότι οι συχνότητες του σήματος αλλάζουν με τον χρόνο και έτσι αν ο μετασχηματισμός Fourier του επόμενου βήματος γινόταν σε όλο το σήμα, τότε θα χανόταν η συχνοτική πληροφορία του. Έτσι, θεωρώντας ότι οι συχνότητες είναι στατικές για ένα μικρό χρονικό διάστημα, μπορούμε να δημιουργήσουμε πλαίσια από το αρχικό σήμα και να εκτελεστεί ο μετασχηματισμός Fourier σε αυτά, ούτως ώστε να εξάγουμε την απαιτούμε συχνοτική πληροφορία.

- ***Παραθυρωποίηση (Windowing)***
  Σε αυτό το βήμα το κάθε πλαίσιο περνά από ένα παράθυρο ούτως ώστε να γίνει ένα smoothing του σήματος και να κρατηθεί η σημαντική περιοχή ενδιαφέροντος. Το πιο ευρέως χρησιμοποιούμενο παράθυρο στην επεξεργασία γλώσσας και φωνής είναι το Hamming, διότι παρέχει μία αποδεκτή ισορροπία μεταξύ συχνοτικής ανάλυσης και δυναμικού εύρους, παρέχοντας μικρότερο σφάλμα σε σχέση με άλλα παράθυρα. Τέλος, επισημαίνουμε, ότι δεν υπάρχει κάποιο συγκεκριμένος κανόνας που να υποχρεώνει την χρήση του Hamming παραθύρου. Αναφορικά, ένα ακόμα παράθυρο που θα μπορούσε να χρησιμοποιηθεί είναι το Hanning.
  Ενδεικτικά αναφέρουμε την μαθηματική έκφραση του Hamming παραθύρου:
  $$w(n) = 0.54 - 0.46 * \cos\left(\frac{2\pi n}{N}\right)$$, όπου 0 ≤ n ≤ N

- ***Μετασχηματισμός Fourier - DFT Spectrum***
  Ύστερα κάνουμε έναν διακριτό μετασχηματισμό Fourier (DFT - Discrete Fourier Transform) σε κάθε ένα από τα πλαίσια που έχουν δημιουργηθεί και λαμβάνουμε το Spectrum του σήματος.
  Η μαθηματική περιγραφή της διαδικασίας αυτής είμαι:
  $$Y_i(k) = \sum_{n=0}^{N-1} y_i(n) * w(n) * e^{-j\frac{2\pi kn}{N}}$$, για k = 0,1,...,N-1 και για κάθε πλαίσιο i

- ***Filter Banks - Mel Spectrum***
  Σε αυτό το βήμα, γίνεται η εξαγωγή των filter banks, εφαρμόζοντας τριγωνικά παράθυρα σε mel - κλίμακα στο Power Spectrum. Η κλίμακα mel χρησιμοποιείται έτσι ώστε να γίνει προσέγγιση του ακουστικού μοντέλου του ήχου όπως τον αντιλαμβάνεται ο άνθρωπος. δηλαδή οι μεταβολές στις χαμηλές συχνότητες να είναι πιο ευδιάκριτες σε σχέση με όμοιες μεταβολές στις υψηλές συχνότητες.
  Η μαθηματική έκφραση για την μετατροπή από Hertz σε mel και αντίστροφα είναι η ακόλουθη:
  $$m = 2595 * \log_{10}\left(1 + \frac{f}{700}\right)\ [mel] \quad και \quad f = 700(10^{\frac{m}{2595}} - 1)\ [Hz]$$

Ενώ η μαθηματική έκφραση των mel filter banks είναι:

$$H_m(k) = \begin{cases} 0 & , \quad k < f(m-1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)} & , \quad f(m-1) \le k < f(m) \\ 1 & , \quad k = f(m) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)} & , \quad f(m) < k \le f(m+1) \\ 0 & , \quad k > f(m+1) \end{cases}$$

Έτσι η περιγραφή του Mel Spectrum είναι η ακόλουθη:

$$s(m) = \sum_{h=f_{k-1}+1}^{f_{k+1}-1} H_m(k) * |Y_i(k)|^2$$

- ***MFCCs - DCT***

  Αποδεικνύεται ότι τα filter banks που προκύπτουν στο παραπάνω βήμα είναι συσχετισμένα, γεγονός που δεν τα καθιστά επιθυμητά για εφαρμογές μηχανικής μάθησης. Για αυτόν τον λόγο, είναι αναγκαία η εξαγωγή των ασυσχέτιστων mfccs. Για τον υπολογισμό των mfccs πραγματοποιείται ένας διακριτός μετασχηματισμός συνημιτόνου (DCT - Discrete Cosine Transform), που προκαλεί την απόσυσχετιση των συντελεστών αυτών.
  Η μαθηματική περιγραφή των mfccs είναι η ακόλουθη:

$$c(n) = \sum_{m=0}^{M-1} \log_{10}(s(m)) * \cos\left(\frac{\pi n(m - 0.5)}{M}\right)$$

  όπου n = 0,1 ,2, ... , C , με C το πλήθος των mfccs.

- ***Κανονικοποίηση***

  Όπως αναφέρθηκε και στην συνοπτική περιγραφή, πολλές φορές για να γίνει ομαλοποίηση του spectrum και να μειωθεί ο θόρυβος και κατ΄ επέκταση να αυξηθεί το SNR, πραγματοποιείται μία κανονικοποίηση των εξαγόντων αποτελεσμάτων.

Τέλος, αναφέρουμε τα πλεονεκτήματα, καθώς και κάποια προβλήματα που έχουμε να αντιμετωπίσουμε επιλέγοντας τα mfccs.

Πλεονεκτήματα:

1. Προκαλείται η κβαντοποίηση του Spectrum, κάνοντας έτσι ευδιάκριτες τις συχνότητες ενδιαφέροντος. Επίσης, συγκρατείται μόνο η σημαντική πληροφορία και απορρίπτεται η πλεονάζουσα, κάνοντας το μοντέλο μας πιο αξιόπιστο.
2. Η υπολογιστική πολυπλοκότητα βρίσκεται σε φυσιολογικά επίπεδα, δίχως να προκύπτουν προβλήματα σχετικά με την αποτελεσματικότητα του αλγορίθμου υπολογισμού των συντελεστών.
3. Τα αποτελέσματα που εξάγονται είναι ορθά και έτσι καταφέρνουμε να αξιολογήσουμε αληθώς το μοντέλο μας

Μειονεκτήματα:

1. Οι συντελεστές mfccs είναι ευάλωτοι στο θόρυβο, συνεπώς όταν η ανάλυση γίνεται παρουσία θορύβου να μην έχουμε ικανοποιητικά αποτελέσματα.
2. Οι συντελεστές αυτοί αν και εξάγουν αρκετά ικανοποιητικά αποτελέσματα κατά την ανάλυση, αντιμετωπίζουν σημαντικά προβλήματα στην διαδικασία της σύνθεσης.

❖ **Γλωσσικά Μοντέλα**: Τα γλωσσικά μοντέλα είναι η βάση των περισσότερων εφαρμογών επεξεργασίας φυσικών γλωσσών (NLP - Natural Language Processing), τεχνητής νοημοσύνης (AI - Artificial Intelligence) και αναζήτησης μηχανικής μάθησης (machine learning research). Χρησιμοποιούνται ευρέως σε εφαρμογές μηχανικής μετάφρασης (machine-translation) και αυτόματης αναγνώρισης φωνής (ASR).
Τα γλωσσικά μοντέλα χωρίζονται σε δύο βασικές κατηγορίες, στα στατιστικά και στα νευρωνικά. Αναλυτικά για κάθε από αυτά παρατίθενται πληροφορίες παρακάτω:

- ***Στατιστικά Γλωσσικά Μοντέλα (Statistical Language Model)***
  Τα στατιστικά γλωσσικά μοντέλα είναι μια κατανομή πιθανότητας σε ακολουθίες λέξεων. Έτσι για μία δεδομένη ακολουθία λέξεων $W=w_1w_2...w_N$ προσδιορίζεται η πιθανότητα $P(W)=P(w_1w_2...w_N)$. Το μοντέλο αυτό παρέχει πληροφορίες για την αναγνώριση λέξεων και εκφράσεων που έχουν παρόμοια ακουστική έκφραση. Πέρα από αυτό, ένα γλωσσικό μοντέλο υπολογίζει όλες τις πιθανότητες οι δύο ξεχωριστές λέξεις να βρίσκονται η μία κοντά στην άλλη. Πέρα από λέξεις, ανάλογα με την εφαρμογή τα γλωσσικά μοντέλα μπορούν να χρησιμοποιηθούν για την αναγνώριση εκφράσεων, συνολικά, ή και μεμονωμένων φωνημάτων.
  Για την μοντελοποίηση αυτών χρησιμοποιούνται κατά κύριο λόγο τα N-grams. Η πιο απλοϊκή προσέγγιση είναι τα unigrams, όπου εκεί η κάθε λέξη θεωρείται σαν ξεχωριστή κατάσταση, δηλαδή κάθε λέξη είναι ανεξάρτητη των προηγούμενων και των επόμενων λέξεων στην ακολουθία εμφάνισης τους. Έτσι ο υπολογισμός της πιθανότητας γίνεται με τον ακόλουθο τύπο:

  $$P(W) = P(w_1 w_2 \dots w_N) = P(w_1)P(w_2) \dots P(w_N)$$

  Για να επιτύχουμε μεγαλύτερη ακρίβεια στην εξαγόμενες πιθανότητες μπορούμε να χρησιμοποιήσομε N-grams μεγαλύτερων τάξεων, όπου εκεί θεωρούμε ότι η κάθε λέξη εξαρτάται από τις N-1 προηγούμενες. Η μαθηματική περιγραφή των N-grams ακλουθεί τον κανόνα της αλυσίδας (chain rule) και είναι η ακόλουθη:
  - $P(W) = P(w_1 w_2 \dots w_N) = \prod_{i=1}^{N} P(w_i|w_1 \dots w_{i-1}) \approx \prod_{i=1}^{N} P\left(w_i|w_{i-(N-1)} \dots w_{i-1}\right)$
  - $P\left(w_i|w_{i-(N-1)} \dots w_{i-1}\right) = \frac{count(w_{i-(N-1)} \dots w_{i-1} w_i)}{count(w_{i-(N-1)} \dots w_{i-1})}$

  Με αυτόν τον τρόπο υπολογίζουμε τις πιθανότητες των συνδυασμών των λέξεων που εμφανίζονται στο corpus που μας δίνεται. Αναφέρεται ότι για τις λέξεις που δεν έχουμε συναντήσει στο corpus, υπάρχουν τρόποι μοντελοποίησης όπου "συνυπολογίζεται" η άγνοια του μοντέλου μας. Μερικές κλασσικές μέθοδοι είναι η backoff και η add-N-smooothing, όπως συναντήθηκαν και στην 1η εργαστηριακή άσκηση.

- ***Νευρωνικά Γλωσσικά Μοντέλα (Statistical Language Model)***
  Τα νευρωνικά γλωσσικά μοντέλα χρησιμοποιούν συνεχείς ενσωματώσεις λέξεων για τον υπολογισμό των προβλέψεων που κάνουν. Τα μοντέλα αυτά, χρησιμοποιούν νευρωνικά δίκτυα και μέσω αυτών γίνεται η εκπαίδευση και η αξιολόγησή τους. Τα νευρωνικά δίκτυα δίνουν την δυνατότητα ενσωμάτωσης των πιθανοτήτων εμφάνισης μιας λέξης η μιας αλληλουχίας λέξεων σαν βάρη με αποτέλεσμα να μην δημιουργούνται τα προβλήματα που θα υπήρχαν στα απλά στατιστικά μοντέλα λόγω του τεράστιου πλήθους των λέξεων εμφάνισης. Έτσι γίνεται η εκπαίδευση και η εξαγωγή των στατιστικών αποτελεσμάτων.

Για την εκπαίδευση των νευρωνικών δικτύων υπάρχουν διάφοροι τρόποι. Ωστόσο, αυτοί που έχουν επικρατήσει είναι οι ακόλουθοι:

(Stochastic) Gradient-Discent

Ο αλγόριθμος stochastic gradient discent περιλαμβάνει την συνεχή επανεκπαίδευση του μοντέλου μας, ανανεώνοντας κάθε φορά τα δοσμένα βάρη, ούτως ώστε να ελαττώνεται το σφάλμα. Η μέθοδος αυτή απαιτεί μια υπερπαράμετρο, τον ρυθμό εκμάθησης (learning rate), όπου πρέπει να ρυθμίζεται από 0.01 - 0.1 ώστε να εξάγονται τα βέλτιστα αποτελέσματα σε μικρό χρονικό διάστημα. Έτσι υπολογίζονται τα βάρη του νευρωνικού δικτύου, δηλαδή οι ζητούμενες πιθανότητες.

CBOW (Continuous Bag Of Words)

Η μέθοδος αυτή περιλαμβάνει την χρήση τόσο επόμενων όσο και προηγούμενων λέξεων, ούτως ώστε να εκτιμήσει τις πιθανότητες εμφάνισης των αλληλουχιών των λέξεων. Η ονομασία προκύπτει, από την συνένωση του continuous, που προέρχεται από την συνεχή εκπαίδευση του νευρωνικού δικτύου και του bag-of-words, διότι η μέθοδος αντιμετωπίζει το μοντέλο μας σαν unigram και η κάθε λέξη θεωρείται ανεξάρτητη από όλες τις άλλες.

Skip-gram

Η μέθοδος αυτή περιλαμβάνει την εξαγωγή της μέγιστης μέσης πιθανότητας εμφάνισης μιας λέξης μέσω της συνεχής επανεκπαίδευσης του μοντέλου μας. Δοσμένης μια ακολουθίας λέξεων το νευρωνικό καταφέρει με συνεχές feedback να προσεγγίσει ακριβέστερη τιμή στην πραγματική για την πιθανότητα εμφάνισης μια συγκεκριμένης λέξης. Αν και είναι πιο αργή σε σχέση με την CBOW , τα αποτελέσματα που εξάγει είναι ακριβέστερα.

- **Ακουστικά Μοντέλα**: Τα ακουστικά μοντέλα χρησιμοποιούνται για την στατιστική αναπαράσταση των ακουστικών πληροφοριών των φωνημάτων, όπου με τον όρο φώνημα περιγράφουμε την βασική μονάδα αναπαράστασης του ήχου. Τα ακουστικά μοντέλα εκπαιδεύονται από έγραφες ήχου και τα αντίστοιχα πρακτικά αυτών των ηχογραφήσεων και μέσω λογισμικού εξάγονται αναπαραστάσεις των ήχων αυτών.
  Οι περισσότερες μηχανές αναγνώρισης φωνής χρειάζονται δύο βασικά εργαλεία, ένα ακουστικό μοντέλο από ηχογραφήσεις και ένα γλωσσικό μοντέλο που δίνει της πιθανότητες εμφανίσεις των λέξεων. Με αυτά τα δύο μοντέλα μπορεί να εξαχθεί από το πρώτο η αναπαράσταση των διάφορων ήχων και σε συνδυασμό με τις στατιστικές πληροφορίες που δίνονται από το δεύτερο να γίνει η αναγνώριση φωνής.

ΒΗΜΑΤΑ ΠΡΟΠΑΡΑΣΚΕΥΗΣ

Βήματα 3.1,3.2,3.3

Εγκαθιστούμε όλα τα απαιτούμενα εργαλεία σύμφωνα με τις οδηγίες που δίνονται και κατεβάζουμε τα αρχεία που χρειάζονται για την συνέχεια της εργαστηριακής άσκησης.

Βήμα 3.4 – Κατασκευή Αρχικού Σκελετού

Αρχικά δημιουργούνται τα κατάλληλα directories , όπως αναφέρονται στις άσκηση. Στα παραπάνω, με τη χρήση του αρχείου “create_env.sh”, γίνεται η προσπέλαση του dataset από το drive και το extraction σε ένα φάκελο “slp_lab2_data”. Επίσης, δημιουργείται το directory "usc/scripts", όπου αποθηκεύονται όλα τα αρχεία “*.py”.

Στην συνέχεια, με το αρχείο "step_3.py", όπου είναι γραμμένο σε γλώσσα Python, δημιουργούμε τα αρχεία "uttids", "utt2spk", "wav.scp" και "text" με τον τρόπο όπου αναγράφεται στην εκφώνηση. Συγκεκριμένα, για κάθε set αντικειμένων που εργαζόμαστε, δηλαδή για τα training, validation και test set δημιουργούνται τα παραπάνω αρχεία και αποθηκεύονται στα αντίστοιχα directories, που έχουν δημιουργηθεί, δηλαδή στα 'data/train', 'data/dev' και 'data/train'. Παρακάτω παρουσιάζεται ένα παράδειγμα για την κατανόηση της αναπαράστασης των δεδομένων:

Αρχείο	Παράδειγμα
train_utterances.txt	usctimit_ema_f1_017
train/uttids	utterance_f1_017
train/utt2spk	utterance_f1_017 speaker_f1
train/wav.scp	utterance_f1_017 ../slp_lab2_data/wav/f1/usctimit_ema_f1_017.wav
train/text	utterance_f1_017 She wore warm fleecy woolen overalls.

Από τα παραπάνω διαπιστώνουμε ότι το πρόγραμμα μας λειτουργεί ορθά και τα αρχεία δημιουργούνται με την μορφή που υποτάσσεται από την εκφώνηση.

Τέλος, γίνεται η επεξεργασία των προτάσεων. Για αυτόν τον σκοπό, μέσω του προγράμματος "step_4.py" δημιουργείται ένα dictionary όπου περιέχει σαν keys τις λέξεις και σαν values τα phonetics της καθεμιάς λέξης. Έτσι κάθε αρχείο "text" ανανεώνεται και αντί για την πρόταση περιέχει τα phonetics αυτής. Για εποπτεία παρατίθεται ένα παράδειγμα:

Αρχείο	Παράδειγμα
train/text	utterance_f1_017 She wore warm fleecy woolen overalls.
train/text (new)	utterance_f1_017 sil sh iy w ao r w ao r m f l iy s iy w uh l ah n ow v er ao l z sil

Από όπου φαίνεται ότι η αντικατάσταση γίνεται με ορθό τρόπο.

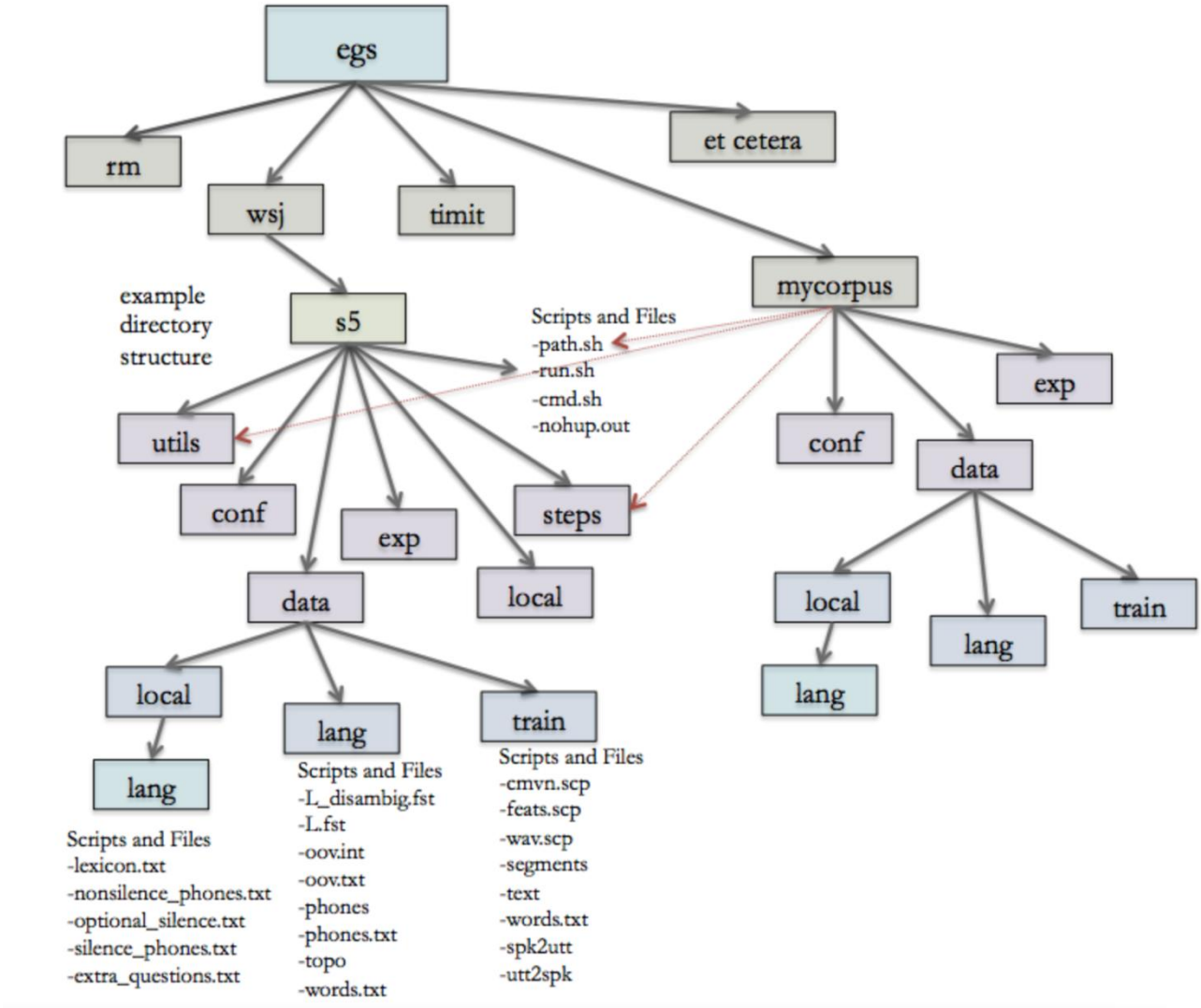
ΒΗΜΑΤΑ ΚΥΡΙΩΣ ΜΕΡΟΥΣ

Βήμα 4.1- Προετοιμασία διαδικασίας αναγνώρισης φωνής για τη USC-TIMIT

Για την κατασκευή του περιβάλλοντος όπου θα εργαστούμε είναι προαπαιτούμενο η δημιουργία συγκεκριμένων subdirectories , καθώς και ορισμένων αρχείων που είναι απαραίτητα για τα επόμενα βήματα. Έτσι κατασκευάζονται τα directories με τον τρόπο που περιγράφεται στην εκφώνηση και δημιουργούνται τα απαιτούμενα soft link με τα αρχεία του kaldi/egs/wsl. Επίσης ένα σημαντικό βήμα είναι η ύπαρξη των αρχείων “path.sh” και “cmd.sh” καθώς και η μετατροπή τους, έτσι ώστε να ανταποκρίνονται στην δομή που χρησιμοποιείται. Τα δύο τελευταία αρχεία είναι απαραίτητα για την υλοποίηση των παρακάτω βημάτων, διότι το αρχείο “cmd.sh” ρυθμίζει τον τρόπο που θα τρέξουν τα scripts στο μηχάνημά μας και το αρχείο “path.sh” είναι απαραίτητο να γίνει source στο directory εργασίας μας ούτως ώστε να μπορούν να «τρέξουν» τα διάφορα έτοιμα scripts. Έχοντας υλοποιήσει όλα τα παραπάνω μπορούμε πλέον να προχωρήσουμε στην δημιουργία των διαφόρων αρχείων που θα χρησιμοποιηθούν για την εκπαίδευση των μοντέλων μας.

Βήμα 4.2 Προετοιμασία γλωσσικού μοντέλου

Για να μπορέσουμε να εξάγουμε τα ακουστικά χαρακτηριστικά και να εκπαιδεύσουμε το μοντέλο μας απαιτούνται κάποια αρχεία που θα αποτελούν το γλωσσικό μας μοντέλο. Όλα τα directories που βρίσκονται στον φάκελο kaldi/egs έχουν μία συγκεκριμένη μορφή η οποία απεικονίζεται παρακάτω:



Όπως φαίνεται και στο παραπάνω σχεδιάσματά, έτσι και στο directory μας υπάρχουν οι φάκελοι conf , data και exp, καθώς και τα subdirectories τους. Στον υποφάκελο data αποθηκεύονται όλα τα λεξιλογικά στοιχεία του ακουστικού μοντέλο μας καθώς και διάφορα αρχεία που δημιουργούνται κατά την δημιουργία του γλωσσικού μοντέλο. Στον υποφάκελο exp βρίσκονται τα αποτελέσματα που εξάγονται κατά την εκπαίδευση και αξιολόγηση του γλωσσικού μοντέλου. Τέλος στον φάκελο conf, περιέχεται ένα μοναδικό αρχείο το “mfcc.conf”, που περιέχει πληροφορίες για την εξαγωγή των mfccs.

Τα βασικά αρχεία που χρησιμοποιούνται για την δημιουργία του γλωσσικού μας μοντέλου και περιέχουν τις πληροφορίες που ζητούνται στην εκφώνηση είναι τα εξής:

- silence_phones.txt
- optional_silence.txt
- nonsilence_phones.txt
- lexicon.txt
- lm_traint.text, lm_dev.text, lm_test.text
- extra_questions.txt

Έχοντας δημιουργήσει αυτά τα αρχεία, μπορούμε πλέον να ξεκινήσουμε την πλήρωση του γλωσσικού μοντέλου. Αρχικά, δημιουργούμε ένα symbolic link με το αρχείο "built-lm.sh" της IRSTLM και μέσω αυτού δημιουργούμε την ενδιάμεση μορφή του γλωσσικού μοντέλου όπου αποθηκεύεται στον φάκελο data/local/lm_tmp. Συγκεκριμένα, εξάγουμε δύο γλωσσικά μοντέλα, ένα unigram και ένα bigram, ώστε να διαπιστώσουμε ποιο από αυτά είναι πιο αξιόπιστο. Στην συνέχεια, δημιουργούμε τα compiled γλωσσικά μοντέλα, ένα για το unigram και ένα το bigram, όπου αποθηκεύονται στον φάκελο data/local/nist_lm σε μορφή ARPA. Έπειτα, κατασκευάζουμε το FST του λεξικού της γλώσσας (L.fst) με την εντολή "prepare_lang.sh" και λαμβάνομαι αποτελέσματα στους υποφακέλους data/local/lang και data/lang, που περιέχονται αρχεία περιγραφής του FST καθώς και τον συμβόλων εισόδου και εξόδου. Ταξινομούμε τα αρχεία "wav.spc", "text" και "utt2spk" των υποφακέλων data/train, data/dev και data/test για να βρίσκονται στην απαιτούμενη μορφή και συγχρόνως δημιουργούμε το αρχείο "spk2utt" για κάθε ένα από τα προηγούμενα sets. Τέλος, μέσω του αρχείου "timit_format_data.sh" που παρέχεται στο github του μαθήματος, δημιουργείται το FST της γραμματικής (G.fst) , ένα για κάθε μοντέλο, unigram και bigram, όπου αποθηκεύονται στους φακέλους data/lang_phones_ug και data/lang_phones_bg αντίστοιχα.

ΕΡΩΤΗΜΑ 1

Για να εξάγουμε ένα πρώτο συμπέρασμα για τα γλωσσικά μοντέλα μας, unigram και bigram, υπολογίζουμε το perplexity (σύγχυση) στο validation και στο test set για καθένα από αυτά.

Αρχικά, αναφέρεται ότι το perplexity είναι μία μετρική που μας δείχνει πόσο καλά μπορεί το μοντέλο μας να προβλέψει την επόμενη λέξη. Δοσμένης μιας ακολουθίας τυχαίων λέξεων $W = w_1 w_2 ... w_N$, η μετρική αυτή υπολογίζεται από τον ακόλουθο τύπο:

$$PP(W) = P(w_1 w_2 \dots w_N)^{-\frac{1}{N}} = \sqrt[N]{\frac{1}{P(w_1 w_2 \dots w_N)}} \xrightarrow{chain\ rule} PP(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i | w_1 \dots w_{i-1})}}$$

Και παίρνοντας την προσέγγιση για bigrams, έχουμε: $PP(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_{i-1})}}$

Από τα παραπάνω προκύπτει ότι όσο πιο ακριβές είναι το μοντέλο μας, τόσο μεγαλύτερες πιθανότητες εξάγονται και συνεπώς το perplexity ελαχιστοποιείται.

Έτσι εξάγουμε τα στατιστικά αποτελέσματα που αναφέραμε μέσω του "compile-lm" και έχουμε:

- Unigram
  - Validation Set: Nw=6357, PP=57.69, PPwp=21.42, Nbo=0, Noov=183, OOV=2.88%
  - Test Set: Nw=6179, PP=57.81, PPwp=22.04, Nbo=0, Noov=184, OOV=2.98%

- Bigram
  - Validation Set: Nw=6357, PP=38.91, PPwp=14.44, Nbo=365, Noov=183, OOV=2.88%
  - Test Set: Nw=6179, PP=38.84, PPwp=14.81, Nbo=367, Noov=184, OOV=2.98%

Τα παραπάνω αποτελέσματα είναι σχετικά αναμενόμενα, αφού και στις δύο περιπτώσεις βλέπουμε ότι το validation set έχει ελάχιστα καλύτερο perplexity, γεγονός που επαληθεύεται αν παρατηρήσουμε τα αρχεία των τριών sets. Το train set, βλέπουμε ότι περιέχει σχεδόν όλες τις προτάσεις των ηχογραφήσεων και τα άλλα δύο set έχουν σχεδόν παρόμοιο πλήθος. Έτσι τα αποτελέσματα που λαμβάνονται είναι σχεδόν παρόμοια για τα δύο sets με το validation σετ να έχει ελάχιστα περισσότερες κοινές προτάσεις με το train set. Ακόμα, συγκρίνοντας τα αποτελέσματα των δύο μοντέλων, είναι εμφανές ότι το μικρότερο perplexity λαμβάνεται στην περίπτωση του bigram, όπου είναι αναμενόμενο αφού σε αντίθεση με το unigram η κάθε λέξη δεν θεωρείται ανεξάρτητη και υπάρχει σχέση μεταξύ προηγούμενων και επόμενων λέξεων.

Βήμα 4.3 Εξαγωγή ακουστικών χαρακτηριστικών

Σε αυτό το βήμα, γίνεται η εξαγωγή των ακουστικών χαρακτηριστικών μέσα από το γλωσσικό μοντέλο που δημιουργήθηκε στο προηγούμενο βήμα. Συγκεκριμένα, εξάγονται οι συντελεστές mfccs μέσω του αρχείου "make_mfcc.sh" και υπολογίζεται το cepstral mean and variance normalization μέσω του αρχείου "compute_cmvn_stats.sh". Και οι δύο αυτές εντολές δημιουργούν αρχεία, τα αποθηκεύονται στους υποφακέλους των τριών sets.

ΕΡΩΤΗΜΑ 2

Όπως έχουμε ήδη αναφέρει οι mfccs είναι ευάλωτοι στον θόρυβο. Η διαδικασία cmvn εξυπηρετεί ακριβώς αυτών τον σκοπό. Μέσω αυτής, κάποια χαρακτηριστικά των συντελεστών mfccs γίνονται ανθεκτικά στον θόρυβο έτσι ώστε να λαμβάνουμε συνεκτικά αποτελέσματα ακόμα και από θορυβώδη κανάλια. Η διαδικασία cmvn στοχεύει στην γραμμική μετατροπή των cepstral τροχιών έτσι ώστε να επιτυγχάνεται μηδενική μέση τιμή και μοναδιαία διακύμανση. Παρακάτω παρουσιάζεται αναλυτικά η διαδικασία:

- CMN – Cepstal Mean Normalization

  Η διαδικασία CMN ή CMS (Cepstal Mean Subtraction) χρησιμοποιείται ευρέως στις εφαρμογές ASR συστημάτων και στοχεύει στην μείωση των διαστρεβλώσεων που εισάγονται από την χρήση διαφορετικών επικοινωνιακών καναλιών ή συσκευών μαγνητοσκόπησης και στην μείωση του θορύβου που τις συνοδεύουν. Η μαθηματική περιγραφή της διαδικασία αυτής είναι η ακόλουθη:

  $$c_{n,t}^{CMN} = c_{n,t} - \mu = c_{n,t} - \frac{1}{T}\sum_{t=1}^{T} c_{n,t}$$

  όπου n αναπαριστά την n-ιοστή cepstal διάσταση και t το cepstal δείγμα στο παρόν παράθυρο.

- CVN – Cepstal Varience Normalization

  Μια διαδικασία που συναντάται ως συμπληρωματική της CMN είναι η CVN, που εκτιμά την διακύμανση, σ_n , για κάθε cepstal διάσταση έτσι ώστε να γίνεται μοναδιαία. Η μαθηματική περιγραφή της είναι:

  $$c_{n,t}^{CVN} = \frac{c_{n,t}}{\sigma_n} = \frac{c_{n,t}}{\sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(c_{n,t} - \mu\right)^2}}$$

  όπου n αναπαριστά την n-ιοστή cepstal διάσταση και t το cepstal δείγμα στο παρόν παράθυρο.

- CMVN - Cepstal Mean and Varience Normalization

  Συνδυάζοντας τις διαδικασίες CMN και CVN επιτυγχάνεται η διαδικασία CMVN, που στοχεύει στο να κάνει τους υπολογιζόμενους συντελεστές ανθεκτικούς στον θόρυβο, κλιμακώνοντας και περιορίζοντας το εύρος της διακύμανσης των χαρακτηριστικών του cepstum. Αξιοποιώντας τους παραπάνω τύπους μπορούμε να εξάγουμε την μαθηματική έκφραση της διαδικασίας ως ακολούθως:

  $$c_{n,t}^{CMVN} = \frac{c_{n,t}^{CMN}}{\sigma_n}$$

ΕΡΩΤΗΜΑ 3

Για να εξάγουμε τα χαρακτηριστικά που ζητούνται, θα χρησιμοποιήσουμε τις εντολές “feat-to-len” και “feat-to-dim” του kaldi. Για τον υπολογισμό των frames που εξάγωνται εντελούμαι την εντολή “feat-to-len scp:data/train/feats.scp ark,t:data/train/feats.lengths” και έτσι δημιουργέιται το αρχείο “data/train/feats.lengths”, όπου σε αυτό αποθηκεύονται το πλήθος των εξαγόμενων frames για κάθε πρόταση του train set. Για τις πρώτες πέντε προτάσεις έχουμε τα ακόλουθα αποτελέσματα:

1. **usctimit_ema_f1_001 237**
2. **usctimit_ema_f1_002 377**
3. **usctimit_ema_f1_003 317**
4. **usctimit_ema_f1_005 399**
5. **usctimit_ema_f1_006 338**

Για τον υπολογισμό των διαστάσεων των χαρακτηριστίκών εκτελούμε την εντολή ”feat-to-dim scp:data/train/feats.scp -”, από όπου προκύπτει ότι **η διάσταση των χαρακτηριστικών του train set είναι 13**.

Βήμα 4.4 Εκπαίδευση ακουστικών μοντέλων και αποκωδικοποίηση προτάσεων

Βήμα 4.4.1:

Χρησιμοποιώντας την εντολή του Kaldi: “steps/train_mono.sh data/train data/lang exp/mono”, δημιουργούμε το monophone GMM-HMM ακουστικό μοντέλο, το οποίο ζητείται.

Τα ορίσματα είναι τα εξής:

1. data/train: Το directory, όπου βρίσκονται τα train δεδομένα, στα οποία θέλουμε να εκπαιδευτεί το μοντέλο μας.
2. data/lang: Το directory, όπου βρίσκεται το λεξικό, το οποίο χρησιμοποιούμε ώστε να εκπαιδευτεί το μοντέλο μας.
3. exp/mono: Το directory, όπου θα αποθηκευτεί το μονοφωνικό μοντέλο.

Πρακτικά σε αυτό το βήμα, έχουμε δημιουργήσει το HC κομμάτι του γράφου αναγνώρισης φωνής.

Βήμα 4.4.2:

Δημιουργούμε το γράφο HCLG, για καθένα από τα γλωσσικά μοντέλα μας(unigram, bigram), σύμφωνα με την αντίστοιχη γραμματική “G” και το κοινό λεξικό “L”. Η αντίστοιχες εντολές είναι οι παρακάτω:

“utils/mkgraph.sh data/lang_phones_ug exp/mono exp/mono/ug_graph”
“utils/mkgraph.sh data/lang_phones_bg exp/mono exp/mono/bg_graph”

Τα “argumetns “ είναι τα εξής:

1. data/lang_phones_bg/ug: Το dir, όπου βρίσκεται το G-fst (το οποίο αντιστοιχεί στο κάθε μοντέλο) και το λεξικό, στο οποίο βασιζόμαστε.
2. exp/mono: Το dir του ακουστικού μέρους το συνολικού γράφου, που δημιουργήσαμε στο προηγούμενο βήμα.
3. exp/mono/ug-bg_graph:

4. το dir, όπου θα αποθηκευτεί η σύνθεση των παραπάνω αυτομάτων. (HCLG.fst)

Βήματα 4.4.3, 4.4.4:
Εκτελούμε αποκωδικοποίηση του γράφου μας, με τον αλγόριθμο Viterbi (ένα DP συνδυαστικό αλγόριθμο μεγιστοποίησης πιθανότητας), μέσω των εντολών:

"steps/decode.sh exp/mono/ug_graph data/dev exp/mono/decode_dev_ug"
"steps/decode.sh exp/mono/ug_graph data/test exp/mono/decode_test_ug"
"steps/decode.sh exp/mono/bg_graph data/dev exp/mono/decode_dev_bg"
"steps/decode.sh exp/mono/bg_graph data/test exp/mono/decode_test_bg"

Τα ορίσματα είναι τα εξής:

1. exp/mono/ug-bg_graph: Directory στο οποίο είναι αποθηκευμένος ο γράφος που αντιστοιχεί στο εκάστοτε n-gram
2. data/dev-test: Dataset για το οποίο τρέχει ο παραπάνω αλγόριθμος. Δηλαδή για δεδομένη ακολουθία παρατηρήσεων (ήχων -> MFCCs) βρίσκουμε την πιθανότερη ακολουθία φωνημάτων.
3. exp/mono/decode_test-dev_bg-ug: Directory αποθήκευσης αποκωδικοποίησης και στατιστικών

Ενδεικτικά αναφέρουμε τα παρακάτω στατιστικά αποτελέσματα των 2 μονοφωνικών μοντέλων στο κάθε dataset :

HCLG-Graph	Dataset	Insertions	Deletions	Substitutions	Correct Words	Ratio	PER
G-Unigram	dev	79	1688	1399	6174	3166 / 6174	51.28
G-Unigram	test	66	1582	1308	5995	2956 / 5995	49.31
G-Bigram	dev	115	1248	1501	6174	2864 / 6174	46.39
G-Bigram	test	107	1134	1375	5995	2616 / 5995	43.64

Πίνακας 1

Παρατήρηση:
Είναι εμφανής η βελτίωση των PER, από unigram σε bigram, όπως και ήταν αναμενόμενο, λόγω της μεγαλύτερης ακρίβειας και πληροφορίας που διαχειρίζονται τα bigrams, η οποία οφείλεται στον τρόπο μοντελοποίησης του κειμένου (2-αδες λέξεων/φωνημάτων).

Σχόλια:

- Για την αξιολόγηση του παραπάνω "decoding" των μοντέλων μας, χρησιμοποιείται η μετρική :

  PER(WER) = (S+D+I)/N

  όπου:

  S: αριθμός των substitutions
  D: αριθμός των deletions
  I: αριθμός των insertions
  N: αριθμός των ορθών λέξεων

Πρακτικά αυτό, το παραπάνω error rate, ποσοτικοποιεί την απόδοση του ακουστικού μας μοντέλου. Δεδομένου ενός dataset από "ήχο" (στην προκειμένη περίπτωση αρχεία *.wav) αλλά και των αντίστοιχων transcription, συγκρίνεται και υπολογίζεται μέσω της Levenshtein Distance (με ένα DP αλγόριθμο σύγκρισης συμβολοσειρών) λεξικογραφική απόσταση των δύο λέξεων. Δηλαδή υπολογίζει το είδος και το πλήθος των παραπάνω λειτουργιών που έχουν λάβει χώρα κατά τη διάρκεια του decoding και κανονικοποιεί με το πλήθος των σωστών λέξεων, οι οποίες βρίσκονται στο reference transcription. Επομένως, γίνεται αντιληπτό ότι, όσο μικρότερο το WER(PER), τόσο καλύτερη απόδοση έχει το μοντέλο μας. (Σημείωση: Αναφερόμαστε σε PER καθώς γίνεται ανάλυση σε επίπεδο φωνήματος).

- Αξίζει να σημειωθεί ότι, ενώ το WER είναι ένας σημαντικός παράγοντας για την αξιολόγηση (ή επιλογή αν πρόκειται για χρήστες) των ASR συστημάτων δεν προσδιορίζει επακριβώς την αποδοτικότητα αυτών, καθώς υπάρχουν πολλοί παράγοντες, οι οποίοι μπορούν να την επηρεάσουν εκτός της ποιότητας των μοντέλων. Ακολούθως, παρατίθενται κάποιοι από αυτούς:
  - Ποιότητα ηχογράφησης
  - Ποιότητα του μικροφώνου
  - Προφορά του ομιλητή
  - Εξειδικευμένοι τεχνολογικοί όροι, πόλεις, ονόματα κλπ
  - Θόρυβος του περιβάλλοντος ηχογράφησης

Παρατήρηση:
Για την εκπαίδευση του μονοφωνικού μοντέλου(G-bigram/unigram) ακολουθείται ο παρακάτω αλγόριθμος:

1. Train monophone using EM
2. Find alingments by Viterbi algorithm
3. Go to 1. using alignments until minimization of neagative log-likelihood

Η εκπαίδευση ενός ASR μοντέλου είναι ιεραρχική, εκπαιδεύοντας συνεχώς πιο πολύπλοκα μοντέλα χρησιμοποιώντας τα προηγούμενα. Μας δίνεται ένα dataset και ένα referecne transcription, χωρίς όμως τα timestamps των αντίστοιχων φωνημάτων, το οποίο είναι απαραίτητο διότι στα αρχεία ήχου μπορεί να υπάρχει θόρυβος σε κάποιο χρονικό διάστημα ή σιωπή και θέλουμε να διακρίνουμε τις χρονικές στιγμές, που εκφέρεται ένα φώνημα λέξη.
Επομένως, η εκπαίδευση του μονοφωνικού μοντέλου γίνεται "iteratively", χρησιμοποιώντας τον αλγόριθμο EM και τα alignmetns που έχουν προκύψει.

<u>Decoding:</u>
Αρχικά, οφείλουμε να σημειώσουμε τις παρακάτω υποθέσεις των Hidden Markovian Models:

1. Μία ακουστική παρατήρηση x είναι ανεξάρτητες με τις άλλες, δεδομένης της κατάστασης που την παρήγαγε.
2. Μια κατάσταση είναι ανεξάρτητη από τις υπόλοιπες, δεδομένης της προηγούμενης κατάστασης.

Επομένως, καταλήγουμε στον παρακάτω τύπο υπολογισμού της από κοινού πυκνότητα πιθανότητας:

$$P(X,Q) = P(X/Q)*P(Q) \quad (1)$$

Στο Decoding, καλούμαστε να βρούμε την πιο πιθανή ακολουθία hidden states (όπου ένα σύνολο αυτών αντιστοιχεί σε ένα φώνημα) δεδομένης μιας ακολουθίας παρατηρήσεων. Ουσιαστικά αρκεί να υπολογίσουμε:

$$Q' = \text{argmax}(Q)\ \{P(Q/X)\ \} = \text{argmax}(Q)\{P(Q,X)/P(X)\} = \text{argmax}(Q)\{P(Q,X) \quad (2)$$

Λόγω της σχέσης (1) η (2), γίνεται: Q’ = argmax(Q){P(X/Q)*P(Q)} (3)
όπου στην εξίσωση (3) οι όροι αναλύονται ως εξής:

- **Likelihood(P(X/Q)):** Υπολογίζεται από το ακουστικό μοντέλο GMM-HMM, DNN-HMM(HC-part of the total graph)
- **Prior(P(Q)):** Υπολογίζεται από το γλωσσικό μοντέλο(unigram, bigram/ G-part of the total graph HCLG)

Στο decoding του lattice γράφου, που προκύπτει, καθοριστικό ρόλο έχουν οι 2 υπερπαράμετροι “LMWT”(LMFS), “WIP”.

❖ **Language Model Scaling Factor(LMSF):**
Όπως θα αναλυθεί κι παρακάτω, για τον υπολογισμό των emition probabilities, εκπαιδεύουμε Multivariate Mixture Gaussian Models (GMMs), τα οποία μοντελοποιούν μέσω του clustering, το dataset, κατηγοριοποιώντας κάθε observation σε μια κανονική κατανομή πιθανότητας. Επομένως, υπολογίζουμε την πιθανότητα ενός “acoustic frame”, δεδομένου ενός subphone (hidden state). Με το να εκπαιδεύουμε classifiers, για κάθε ακουστικό frame και να υπολογίζουμε το γινόμενο των πιθανοτήτων για τη συνολική λέξη, δε γίνεται καλή εκτίμηση της πιθανότητας του κάθε subphone, λόγω της συνέχειας των ακουστικών frames. Συνεπώς, αν θέλουμε να λάβουμε υπόψιν μας το “acoustic context”, θα πρέπει να κάνουμε ένα reweight στις πιθανότητες και να αυξήσουμε την πιθανότητα για ένα δεδομένο frame. Αυτό το σκοπό επιτυγχάνει, ο παραπάνω παράγοντας. Οπότε η σχέση (3) γίνεται ως εξής:

$$Q' = \text{argmax}_Q\{P(X/Q)*P(Q)^{LMFS}\} \quad (4)$$

Οι τιμές που παίρνει συνήθως είναι από 5-15. Στην προκειμένη περίπτωση δοκιμάστηκαν οι τιμές 7-17.

❖ **Word insertion Penalty:**
Το παραπάνω scaling, δημιουργεί προβλήματα στο decoding. Το γλωσσικό μοντέλο P(Q), επηρεάζει το insertion penalty. Για παράδειγμα, αν έχουμε ένα unigram μοντέλο με μέγεθος |V|, κάθε λέξη θα έχει πιθανότητα εμφάνισης 1/|V|. Τότε σε μια πρόταση από N λέξεις θα έχουμε δυνητικά ένα συνολικό λάθος εισαγωγής λέξεων N/|V|. Επομένως, αν η πιθανότητα του γλωσσικού αυτού μοντέλου μειωθεί, τόσο θα μικρύνει η πιθανότητα για την δημιουργία μεγάλων λέξεων. Με την εφαρμογή του LMSF, γίνεται μείωση αυτής της πιθανότητας, οπότε αυξάνεται το insertion penalty. Για την εξισορρόπηση αυτού, προσθέτουμε ένα offset και καταλήγουμε:

$$Q' = \text{argmax}_Q\{P(X/Q)*P(Q)^{LMSF} * WIP^N\}, \text{ N:τάξη μοντέλου} \quad (5)$$

Πρακτικά υπολογίζουμε την log-likelihood πιθανότητα, οπότε η (5) γίνεται:

$$Q' = \text{argmax}_Q \{ \log P(X/Q) + LMSF \times \log P(Q) + N \times \log WIP\} \quad (6)$$

Οι τιμές που δοκιμάστηκαν στην προκειμένη περίπτωση είναι 0.0, 0.5, 1.0.

Στο καλύτερο μοντέλο, το οποίο είναι το μονοφωνικό μοντέλο για “bigram” όπως φαίνεται κι από τον “Πίνακα 1”, οι υπερπαράμετροι είναι οι εξής:

Dataset	LMSF	WIP
dev	7	0.0
test	7	0.0

Πίνακας 2

Βήμα 4.4.5:
Για την εξαγωγή των alignments και η εκπαίδευση του τριφωνικού μοντέλου, χρησιμοποιήθηκαν οι παρακάτω εντολές:

"steps/align_si.sh data/train data/lang exp/mono exp/mono_ali"
"steps/train_deltas.sh 2000 10000 data/train data/lang exp/mono_ali exp/tri"

όπου:

- Data/train: dir του dataset που θέλουμε να εκτιμήσουμε τα επιθυμητά timestamps
- Data/lang: dir του λεξικού
- exp/mono: dir του μονοφωνικού μοντέλου
- exp/mono_ali: dir αποθήκευσης των alginments

Για την εντολή εκπαίδευσης:

- num-leaves: ή αλλιώς HMM states. Ο αριθμός των nodes στο decision tree. Λόγω της ύπαρξης των αλλοφώνων αλλά και την αναπαράσταση των φωνημάτων με παραπάνω από μια κατάσταση(3-5 states for 1 phone), ο αριθμός "2000" είναι αρκετά ικανοποιητικός.

- tot-gauss: συνολικός αριθμός των gaussians, που θα εκπαιδευτούν

Σχόλιο:
Ο αριθμός των nodes του μαρκοβιανού μοντέλου διαφέρει ανάλογα την έκταση του dataset, το σκοπό και το πλήθος των φωνημάτων.

Ακολουθώντας τα προηγούμενα βήματα για την εξαγωγή του HCLG γράφου και το decoding, προκύπτουν τα παρακάτω στατιστικά:

HCLG-Graph	Dataset	Insertions	Deletions	Substitutions	Correct Words	Ratio	PER
G-Unigram	dev	286	726	1396	6174	2408 / 6174	39.00
G-Unigram	test	177	780	1301	5995	2258 / 5995	37.66
G-Bigram	dev	270	587	1295	6174	2152 / 6174	34.86
G-Bigram	test	203	514	1206	5995	1923 / 5995	32.08

Πίνακας 3

ΕΡΩΤΗΜΑ 4
Ένα ακουστικό μοντέλο GMM-HMM, αποτελείται από δύο οντότητες. Το μαρκοβιανά μοντέλα και τα Μίγματα Γκαουσιανών κατανομών. Το καθένα επιτελεί το σκοπό του, με απώτερο στόχο την αναγνώριση φωνής.

❖ **GMM:** Τα GMM είναι μίγματα γκαουσιανών pdfs. Ο σκοπός του είναι το classification του ήχου σε clusters. Στην προκειμένη περίπτωση χρησιμοποιούνται multivariate gaussian mixture models. Η μαθηματική σχέση που τα περιγράφει είναι :

$$P(x) = \sum_{k=1}^{m} P(x|z)\,P(z) = \sum_{k=1}^{m} P(z = e_k)\,P(x|z = e_k) = \sum_{k=1}^{m} c_k N(x/\mu_k, \Sigma_k)$$

όπου z , είναι μια latent variable $\in \{e_1 \dots e_m\}, e_1 = [1\ 0\ 0 \dots 0] \dots, e_k = [0\ 0 \dots 1]$.

Δεδομένου ότι, $z = e_k$:

- $X \sim N(\mu_k, \Sigma_k)$, $\acute{o}\pi o\upsilon\ \mu_k \in \{\mu_1 \dots \mu_\mu\} \in R^d\ \epsilon\acute{\iota}\nu\alpha\iota\ \tau o\ \delta\iota\acute{\alpha}\nu\upsilon\sigma\mu\alpha\ \tau\omega\nu\ mean\ values$,
- $\Sigma_k \in \{\Sigma_1 \dots \Sigma_k\} \in R^{dxd}\epsilon\acute{\iota}\nu\alpha\iota\ o\ covariance\ matrix$.

Εδώ η μεταβλητή "z", δηλώνει την κλάση "k" (γκαουσσιανή), στην οποία αναφερόμαστε. Επομένως:

- $P(x|z)$: mixture components, πρακτικά δηλώνει την αντίστοιχη γκαουσσιανή
- $P(z)$: mixing parameters(a_k), παράμετροι που θα υπολογιστούν με τον EM αλγόριθμο, ώστε να ποσοτικοποιηθεί η συνεισφορά κάθε γκαουσσιανής.

Βήματα EM-αλγορίθμου:

a) E-step:
Αρχικοποιώντας τα mean και covariance ακολουθούμε την παρακάτω διαδικασία.
Για κάθε κλάση υπολογίζουμε τις occupation probs (responsbility probs):

$$r_{ik} = \frac{c_k N(x_i / \mu_k, \Sigma_k)}{\sum_j c_j N(x_i / \mu_j, \Sigma_j)}$$

Σχόλιο:
Αναμένεται αν η κλάση "k", προσεγγίζει πολύ καλά το x_i, μεγάλη τιμή στην αντίστοιχη πιθανότητα r_{ik}, που πρακτικά ισούται με την πιθανότητα ένα στοιχείο να ανήκει στην κλάση "k" $P(k|x_i)$.

b) M-step (Maximization):
Για κάθε κλάση "k":

- $m_k = \sum_i r_{ik}$: total responsibity allocated to cluster k
- $c_k = \frac{m_k}{m}$: fraction of the total assigned to k
- $\mu_k = \frac{1}{m} \sum_i r_{ik}\, x_i$: αν η κλάση "k", δεν χαρακτηρίζει με πολύ καλό τρόπο το x_iτο αντίστοιχο r_{ik}θα επηρεάσει λιγότερο την μέση τιμή:

$$\Sigma_k = \frac{1}{m_k} \sum_i r_{ik}\, (x^{(i)} - \mu_k)^T (x^{(i)} - \mu_k)$$

Σε κάθε ανακύκλωση, ο παραπάνω αλγόριθμος μειώνει το negative log-likelihood error, με αποτέλεσμα κάποια στιγμή να συγκλίνει.

$$L = -log(P(x)) = \sum_i log\,(\sum_k c_k\, N(x_i/\mu_k \Sigma_k)$$

Ο παραπάνω αλγόριθμος χρησιμοποιείται για την εκπαίδευση ενός πλήθος GMMs, τα οποία χρησιμεύουν στην διαδικασία ASR, καθώς υπολογίζουν τα emition probabilities.

❖ **HMM:** Η ASR ανάλυση βασίζεται στις παρατηρήσεις(observations), δηλαδή στα "mfccs" του ήχου, ώστε να προβλέψει την πιο πιθανή ακολουθία φωνημάτων. Το μαρκοβιανό μοντέλο έχει ως είσοδο ήχο (mfccs, acoustic frames) και έξοδο φωνήματα (phonemes). Τα Hidden Markov Models, κάνουν τις εξής υποθέσεις, οι οποίες αναφέρθηκαν κι παραπάνω.

1) Μία ακουστική παρατήρηση **x** είναι ανεξάρτητες με τις άλλες, δεδομένης της κατάστασης που την παρήγαγε.
2) Μια κατάσταση είναι ανεξάρτητη από τις υπόλοιπες, δεδομένης της προηγούμενης κατάστασης.

Καταλήγοντας στην μαθηματική σχέση: **P(X,Q) = P(X/Q)*P(Q)**

Τα GMM-HMM, όπως αναφέρθηκαν είναι υπεύθυνα για τον υπολογισμό της πρώτης πιθανότητας, όντας ακουστικό μοντέλο. Η πιθανότητα λόγω των παραπάνω υποθέσεων μετατρέπεται:

$$P(X/Q) = \prod_{i=1}^{t} P(x_i/q_i)$$

αυτές οι πιθανότητες λέγονται emition probabilities και υπολογίζονται μέσω των GMMs, που έχουν εκπαιδευτεί προηγουμένως.

Τα HMM μοντέλα επιτελούν τα παρακάτω tasks:

- Likelihood: $P(X|\lambda), \lambda = \{a_{ij}, b_j()\}$ *παράμετροι του HMM*
  Υπολογίζει την πιθανότητα μιας ακολουθίας παρατηρήσεων να έχει δημιουργηθεί από ένα HMM. Για τον υπολογισμό αυτό, χρησιμοποιείται ο forward αλγόριθμος. Ο αλγόριθμος αυτός, αθροίζει τις πιθανότητες όλων των μονοπατιών από καταστάσεις, που καταλήγουν στην παρατήρηση X. Πρακτικά, υπολογίζει την πιθανότητα $a_t(s_j) = p(x_1 \dots x_t, s_1 \dots s_t|\lambda)$, όπου δηλώνει την πιθανοτητα να έχει δει την ακολουθία $x_1 \dots . x_t$και να χει δημιουργηθεί από την ακολουθία καταστάσεων $s_1 \dots s_t$.

  Ένας εναλλακτικός αλγόριθμος είναι ο VIterbi, ο οποίος επιλέγει την μεγαλύτερη πιθανότητα σε κάθε αναδρομικό βήμα σε αντίθεση με τον forward, που τις αθροίζει.

- Decoding:
  Δεδομένης μιας ακολουθίας παρατηρήσεων, βρίσκει την ακολουθία φωνημάτων με την μεγαλύτερη πιθανότητα. Για το decoding χρησιμοποιείται ο Viterbi.

  Μια προσέγγιση του παραπάνω αλγορίθμου είναι: $V_t(s_j) = max_{\ i}\{ V_{t-1}(s_i) a_{ij} b_j(x_t)\}$
  Πρόκειται για ένα DP αλγόριθμο, όπου για το backtracking χρησιμοποιείται ένας πίνακας κατά τη διάρκεια της αναδρομής.

- Training:
  Για την εκπαίδευση των HMM, χρησιμοποιείται ο αλγόριθμος EM, ή αλλιώς forward-backward.
  Οι backward πιθανότητες υπολογίζονται ως εξής: $\beta_\tau(s_j) = p(x_{t+1}, x_{t+2}, x_t \mid S(t) = s_j, \lambda)$
  Η παραπάνω δηλώνει την πιθανοτήτων μελλοντικών παρατηρήσεων, ότι το HMM είναι στην κατάσταση s_j την στιγμή t.

Με τον υπολογισμό των forward και backward πιθανοτήτων μπορούμε να υπολογίσουμε πλέον την occupation probability ως:

$$\gamma_t(s_j, m) = P(S(t) = s_j \mid X, \lambda(m)) = \frac{1}{a_T(s_E)} \alpha_t(j)\beta_t(j)$$

η οποία αντιστοιχεί στην πιθανότητα εμφάνισης του m-οστού mixture component στην κατάσταση s_jτην στιγμη t δεδομένης της ακολουθίας X. Οι πιθανότητες αυτές είναι τα λεγόμενα alignments και συγκεκριμένα τα component/state-alignments, καθως πρόκειται για μίγμα γκαουσιανών. Εν συνεχεία, τις χρησιμοποιουμε για να κάνουμε re-estimating τις παραμέτρους των GMMs.

$$\hat{\mu}^{(jm)} = \frac{\sum_{t=1}^{T} \gamma_t(s_j,m) x_t}{\sum_{t=1}^{T} \gamma_t(s_j,m)} \quad (1)$$

$$\hat{\Sigma}^{(jm)} = \frac{\sum_{t=1}^{T} \gamma_t(s_j,m)(x_t - \hat{\mu}^{(jm)})(x_t - \hat{\mu}^{(jm)})^T}{\sum_{t=1}^{T} \gamma_t(s_j,m)} \quad (2)$$

$$\hat{c}^{(jm)} = \frac{\sum_{t=1}^{T} \gamma_t(s_j,m)}{\sum_{l=1}^{M} \sum_{t=1}^{T} \gamma_t(s_j,l)} \quad (3), \text{ mixture components}$$

Προφανώς, γίνεται ανανέωση των emtion probs με αυτό τον τρόπο.

Επιπλέον, υπολογίζουμε τις πιθανότητες state occupation probabilities :

$$\xi_t(s_i, s_j) = P(S(t) = s_i, S(t+1) = s_j \mid X, \lambda) = \frac{a_t(s_i)\, a_{ij}\, b_j(x_{t+1})\beta_{t+1}(s_j)}{a_T(S_e)}$$

Τις πιθανότητες αυτές τις χρησιμοποιούμε για να κάνουμε re-estimate τα transitions.

$$\widehat{a_{ij}} = \frac{\sum_{t=1}^{T} \xi_t(s_i,s_j)}{\sum_{k=1}^{N} \sum_{t=1}^{T} \xi_t(s_i,s_j)} \quad (4)$$

Επομένως, η εκπαίδευση του HMM χωρίζεται σε δύο στάδια:

- E-step(estiomation): Εκτίμηση των forward-backward και occupation πιθανοτήτων $a_t(s_j), \beta_t(j), \gamma_t(s_j), \xi_t(s_i, s_j)$
- M-step(Maximization: Επαναυπολογισμός των mean vectors, covariance matrices, των mixture components και των transitions.

Συνοψίζοντας, τα παραπάνω αποτελούν την εκπαίδευση ενός context independent monophone acoustic model, όπου εν τέλει τα GMMs έχουν διαχωριστεί για κάθε subphone. Μια επέκταση της παραπάνω εκπαίδευσης είναι η χρήση αυτών των γκαουσσιανών για την εκπαίδευση τριφωνικών.

Σχόλιο:
Η εκπαίδευση μπορεί να επιτευχθεί και με τη χρήση του αλγορίθμου Viterbi, όπου υπολογίζουμε τα alignments των φωνημάτων με ένα πέρασμα του trainset στον VIterbi.

ΕΡΩΤΗΜΑ 5

Η a-posteriori πιθανότητα σύμφωνα με τον τύπο του Bayes γράφεται ως εξής:

$$P(Q|X) = P(X|Q)\, P(Q)P(X)$$

Σύμφωνα, με την παραπάνω έκφραση η εύρεση της πιο πιθανής ακολουθίας καταστάσεων ώστε να μεγιστοποιείται η P(Q|X), ανάγεται στην μεγιστοποίηση του: $Q = \text{argmax}Q\{P(X|Q)\, P(Q)\}$

Με τη χρήση του αλγόριθμου VIterbi, βρίσκουμε το μονοπάτι που μεγιστοποιεί την πιθανότητα και με backtracking εντοπίζουμε την ακολουθία. Οφείλουμε να σημειώσουμε ότι, ο Viterbi για decoding διαφέρει από αυτόν για training, καθώς δεν γνωρίζουμε την αντιστοιχία λέξης παρατήρησης.

Ο αλγόριθμος Viterbi είναι ένας DP αλγόριθμος, ο οποίος επιλέγει συνεχώς το μονοπάτι με τη μεγαλύτερη πιθανότητα.

ΕΡΩΤΗΜΑ 6

Η δομή του HCLG γράφου του Kaldi είναι η εξής:

- **H-part:** Το H-κομμάτι του γράφου αποτελεί την "μετάφραση" των ηχητικών χαρακτηριστικών σε φωνήματα. Οι έξοδοι του η H γράφου αναπαριστούν context dependent φωνήματα και οι είσοδοι είναι τα trainsition-ids μεταξύ των states, τα οποία κωδικοποιούν τα gaussian pdf-ids ενός ακουστικού frame. Πάνω σ'αυτό εφαρμόζεται ο Viterbi για το decoding, δημιουργώντας ένα lattice γράφο.

- **C-part:** Ο γράφος C καθορίζει την εξάρτηση των λέξεων(φωνημάτων στη συγκεκριμένη περίπτωση). Πιο αναλυτικά μοντελοποιεί κάθε φώνημα σε ένα fst, όπου έχει πιθανότητες μετάβασης προς τα υπόλοιπα φωνήματα. Για παράδειγμα, όταν θα έχουμε δημιουργήσει την συμβολοσειρά "st", είναι πιο πιθανό να ακολουθεί φωνήεν έναντι συμφώνου.

- **L-part:** Αποτελεί το λεξικό του εκάστοτε μοντέλου. Οι έξοδοι του είναι λέξεις κι οι είσοδοι φωνήματα. Στην προκειμένη περίπτωση, πρόκειται για ένα identity graph, καθώς δουλεύουμε σε επίπεδο φωνημάτων.

- **G-part:** Το μέρος αυτό είναι ένας απλός acceptor, με input και output symbols τα ίδια. Ο στόχος του είναι η κωδικοποίηση μιας γραμματικής ή ενός γλωσσικού μοντέλου (στην περίπτωσή μας ένα για unigram, ένα για bigram).

Η σύνθεση όλων αυτών σχηματίζει τον HCLG γράφο. Η μετάβαση από ήχο σε φωνήματα γίνεται στον H, στον C γίνεται ο έλεγχος λογικών αποτελεσμάτων του H και διόρθωση τους λαμβάνοντας υπόψιν τις πιθανότητες μετάβασης, στον L γίνεται η αντιστοίχιση σε λέξεις και στο G γίνεται η αποδοχή ή όχι με βάση κάποια γραμματική.

Βήμα 4.5 Bonus: Μοντέλο DNN-HMM με PyTorch

Βήματα 4.5.1, 4.5.2:

Ακολουθώντας την ίδια διαδικασία με προηγουμένως, εξάγουμε τα alignments του τριφωνικού μοντέλου και για τα 3 dataset και τα απαραίτητα στατιστικά χαρακτηριστικά. Σημειώνουμε ότι, η εκπαίδευση έγινε μόνο για το triphone, το οποίο βασίζεται στο bigram, καθώς αυτό είχε την καλύτερη απόδοση.

Βήμα 4.5.3:

Μελετώντας την κλάση TorchSpeechDataset, καταλήξαμε ότι, αυτό που κάνει είναι να αποθηκεύει τα προηγούμενα alignments σε μορφή χρήσιμη για την εκπαίδευση του νευρωνικού. Πρακτικά, επιστρέφει ένα πίνακα "feats", ο οποίος είναι ένας πίνακας από διανύσματα με mfccs και έναν πίνακα labels, όπου περιέχει τα φωνήματα των αντίστοιχων προηγούμενων ηχητικών χαρακτηριστικών.

Συνεπώς, μπορούμε εύκολα να εκπαιδεύσουμε ένα DNN classifier με feed-forward αρχιτεκτονική και τα χαρακτηριστικά, που υποδεικνύονται.

Βήμα 4.5.4:
Η εκπαίδευση έγινε βασισμένη στον αρχικό βοηθητικό κώδικα. Η κλάση TorchDNN έχει τα παρακάτω χαρακτηριστικά.

Input layer, 2 hidden layers, output layer:

- Input layer/hidden layers: 1) Batch-Normalization1D στην αρχή του layer
  2) Linear Tranforamtion
  3) Dropout Reguralization in order to avoid overfitting.
  4) ReLU, activation function, μη γραμμική συνάρτηση, η οποία αφήνει ανεπηρέαστα τα θετικά στοιχεία και μηδενίζει τα υπόλοιπα

- output layer: Μόνο γραμμικός μετασχηματισμός

Σχόλιο:
Δοκιμάστηκε κι η προσθήκη της συνάρτησης Softmax στο τελευταίο layer, αλλά ο classifier δεν έδινε τόσο καλά αποτελέσματα πάνω στο dev-set. Ενδεικτικά αναφέρουμε, ότι σαν μετρική χρησιμοποιήθηκε το accuracy (ο λόγος σωστών αντιστοιχίσεων προς τα συνολικά), όπου το μοντέλο μας με την προσθήκη του softmax πετύχαινε ένα ποσοστό 20%.

Μετά από πολλές δοκιμές στις υπερπαραμέτρους καταλήξαμε στις παρακάτω τιμές που μας έδωσαν τον καλύτερο μοντέλο:

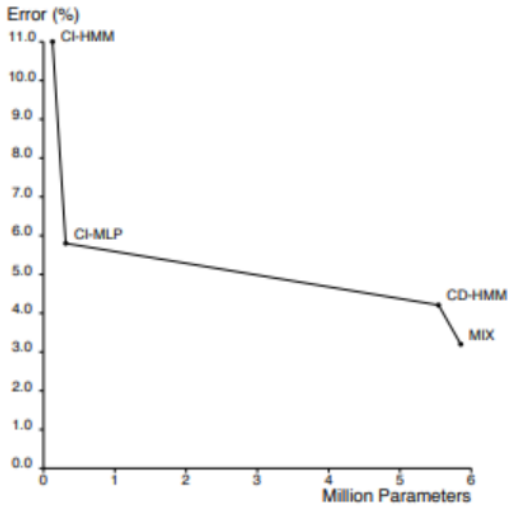
NUM_LAYERS = 2
HIDDEN_DIM = 256
USE_BATCH_NORM = True
DROPOUT_P = .2
EPOCHS = 50
PATIENCE = 3
ETA = 1e-1
BATCH_SIZE=128
AFFINE=True

dev_acc = 0.57
test_acc = 0.58
WER = 29.88
model_name = dnn

Βήμα 4.5.5:
Χρησιμοποιώντας το καλύτερο μας μοντέλο, εξαγάγαμε τις posteriors , με βάση τα οποία έγινε η σχετική αποκωδικοποίηση. Παραθέτουμε παρακάτω τα σχετικά αποτελέσματα.

	Dataset	Insertions	Deletions	Substitutions	Correct Words	Ratio	WER(PER)	LMSF	WIP
DNN-HMM	test	168	554	1063	5974	1785 / 5974	29.88	2	0.5

Σχόλιο:
Είναι φανερή η βελτίωση, αλλά όχι σε πολύ μεγάλο ποσοστό. Θα μπορούσε κανείς να πει ότι είναι λογικό, καθώς παρατηρώντας τον επόμενο διάγραμμα, φαίνεται ότι τα CD μοντέλα δεν έχουν πολύ μεγάλη διαφορά.



Διάγραμμα 1

ΕΡΩΤΗΜΑ 7
Η βασική διαφορά των δύο αυτών μοντέλων κυμαίνεται στην διαδικασία υπολογισμού της a-prosteriori πιθανότητας.

Το GMM το πετυχαίνει, εκπαιδεύοντας gaussians clusters και προσδίδοντας έτσι το emition prob με βάση την pdf, στην οποία αντιστοιχεί το κάθε στοιχείο.
Το DNN, πρόκειται για ένα deep learning classifier, το οποίο εκπαιδεύεται στο ποια είναι πιο πιθανή αντιστοιχία με βάση τα alignments, που έχουν προκύψει από το GMM.

Το DNN υπολογίζει την 'a-posterior' πιθανότητα P(Q|X), η οποία μετατρέπεται στην direct prob ως εξής:

$$P(xi/qi) = P(qi|xi)P(xi)P(qi)$$

Πλεονεκτήματα:(DNN vs GMM):
1. Είναι προσαρμοστικό-ευέλικτο. Λόγω της συνέχειας των ακουστικών frames και του context dependency των λέξεων είναι απαραίτητος ο εντοπισμός των αλληλεξαρτήσεων και των διαφορετικών αλλοφώνων, το οποίο πετυχαίνει ένα DNN.
2. Έχουν λιγότερες παραμέτρους και μπορούν να γενικεύσουν περισσότερο από τα GMMs.
3. Τα DNNs μοντελοποιούν από κοινού κατανομές διαφορετικών κλάσεων (tied learning), ενώ τα GMMs υπολογίζουν την κατανομή για κάθε state ξεχωριστά.
4. Τα DNNs μπορούν να υπολογίσουν τα alignments μοντελοποιώντας μεγάλο πλήθος από frames(συνήθως 40 και πάνω), ενώ τα GMMs μόλις 7-9 λόγω του distribution convexity (οι mixture components αθροίζουν στο 1).

Είναι προφανές ότι, υπάρχει μια αλληλουχία των βημάτων της ASR ανάλυσης. Το ένα επίπεδο τροφοδοτεί το άλλο. Επομένως, δε θα μπορούσαμε να προχωρήσουμε κατευθείαν στην εκπαίδευση του DNN-HMM, καθώς είναι απαραίτητα τα alignments που παρέχουν τα GMMs για την εκπαίδευση του.

Προτάσεις Βελτίωσης:

1. Ως επέκταση αυτού, θα μπορούσαμε να εξάγουμε από το νέο μοντέλο(DNN-HMM) νέα alignments τα οποία θα τροφοδοτήσουμε στο μοντέλο μας για να βελτιωθεί.
2. Χρήση RNNs στη θέση των DNNs. Είναι αποδεδειγμένο ότι, τα RNNs έχουν καλύτερο perforamnce από τα DNNs σε αυτές τις εφαρμογές. Δίκτυα όπως, τα LSTMs (unidirectional/bidirectional).
3. Μεγαλύτερο train-dataset

ΕΡΩΤΗΜΑ 8

Αυτή η τεχνική, χρησιμοποιείται στην εκπαίδευση deep learning nets, για την σταθεροποίηση του ρυθμού μάθησης κι την μείωση των συνολικών εποχών που χρειάζονται. Εφαρμόζεται στην αρχή κάθε layer για την κανονικοποίηση της εισόδου του εκάστοτε miini-batch.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [R&S] Theory and Applications of Digital Speech Processing των Lawrence R. Rabiner and Ronald W. Schafer (Pearson, 2011)
- https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html
- https://maelfabien.github.io/machinelearning/speech_reco_1/#introduction-to-hmm-gmm-acoustic-modeling
- https://eleanorchodroff.com/tutorial/kaldi/index.html
- https://wiki.aalto.fi/display/ITSP/Cepstrum+and+MFCC
- https://www.researchgate.net/publication/221616659_Choice_of_Mel_filter_bank_in_computing_MFCC_of_a_resampled_speech
- https://en.wikipedia.org/wiki/Mel-frequency_cepstrum
- https://en.wikipedia.org/wiki/Language_model
- https://machinelearningmastery.com/statistical-language-modeling-and-neural-language-models/
- https://nlp.stanford.edu/IR-book/html/htmledition/types-of-language-models-1.html
- https://en.wikipedia.org/wiki/Acoustic_model
- https://www.microsoft.com/en-us/research/project/acoustic-modeling/
- https://web.stanford.edu/class/cs124/lec/languagemodeling.pdf
- https://www.researchgate.net/publication/314127935_Comparison_of_Cepstral_Normalization_Techniques_in_Whispered_Speech_Recognition
- https://www.isca-speech.org/archive_open/archive_papers/robust2004/rob4_38.pdf
- http://www.inf.ed.ac.uk/teaching/courses/asr/2019-20/asr08-hybrid_hmm_nn.pdf
- https://www.researchgate.net/publication/259889707_The_INTERSPEECH_2014_computational_paralinguistics_challenge_Cognitive_physical_load
- https://infoscience.epfl.ch/record/213067/files/Himawan_ASRU_2015.pdf
- http://cslt.riit.tsinghua.edu.cn/mediawiki/images/4/4a/Without_gmm.pdf
- https://static.googleusercontent.com/media/research.google.com/el//pubs/archive/43908.pdf

***Υποσημείωση 1**: Τα παραπάνω αποτελέσματα που προκύπτουν από την εκπαίδευση των μοντέλων δεν είναι ντετερμινιστικά και κατ' αυτού, ξανατρέξουμε τον κώδικα θα λάβουμε διαφορετικά στατιστικά αποτελέσματα. Τα αποτελέσματα που αναγράφονται στην αναφορά μας, είναι αυτά που προέκυψαν κατά την εκτέλεση της εργαστηριακής άσκησης, στα τοπικά μας μηχανήματα.

***Υποσημείωση 2**: Στα παραδοθέντα αρχεία συμπεριλαμβάνεται αρχείο "README.txt", όπου παρέχει αναλυτικές οδηγίες για την εκτέλεση των scripts.

ΤΕΛΟΣ 2^{ης} ΕΡΓΑΣΤΗΡΙΑΚΗΣ ΑΝΑΦΟΡΑΣ