Εθνικο Μετσοβιο Πολυτεχνειο

Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων

Πρωτη Εργαστηριακη Άσκηση: Αναγνωριση φωνης με το KALDI Toolkit
Επεξεργασια Φωνης και Φυσικης Γλωσσας

Αναστασία Χριστίνα Λίβα
03119029

Περιεχόμενα

Μέρος 1: Περιγραφή

Σύμφωνα με το αρχείο README, ο φάκελος εργασίας είναι ο «usc». Μετά την εγκατάσταση του KALDI, πρέπει να έχει την παρακάτω δομή:

```
kaldi├──
 egs
        ├── usc
        ├── main.sh
                ├── scripts
                ├── build-lm.sh
                ├── build-lm-util.sh
                ├── compile-lm-util.sh
                ├── clean.sh
                ├── get_deps.sh
                ├── perplexity-util.sh
                ├── prep_lang-util.sh
                ├── part3.sh
                ├── part4-1.sh
                ├── part4-2.sh
                ├── part4-3.sh
                ├── part4-4.sh
```

Τα βήματα της άσκησης υλοποιούνται μέσω των παρακάτω αρχείων κώδικα:

- part3.py
- part4-1.py
- part4-2.py
- part4-3.py
- part4-4.py

Τα υπόλοιπα scripts έχουν γραφτεί για να διευκολύνουν την υλοποίηση με τις ακόλουθες λειτουργίες:

- clean.sh: Καθαρίζει τον φάκελο, διατηρώντας μόνο τα scripts και το main.sh.
- get_deps.sh: Κατεβάζει τα απαραίτητα αρχεία από το Google Drive και τα απαραίτητα scripts από το GitHub του μαθήματος.
- build-lm-util.sh: Κάνει source το path και εκτελεί το build-lm.sh.
- compile-lm-util.sh: Αυτό το script ρυθμίζει το περιβάλλον εκτέλεσης (source το path) και στη συνέχεια εκτελεί το compile-lm.sh.
- prep_lang-util.sh: Αυτό το script ρυθμίζει το περιβάλλον εκτέλεσης (source το path) και στη συνέχεια εκτελεί το prep_lang.sh.

Για να εκτελέσω ολόκληρο τον κώδικα, μπορώ να χρησιμοποιήσω το script main.sh εντός του working directory (usc). Αυτό το συγκεκριμένο script εκτελεί τα παρακάτω βήματα:

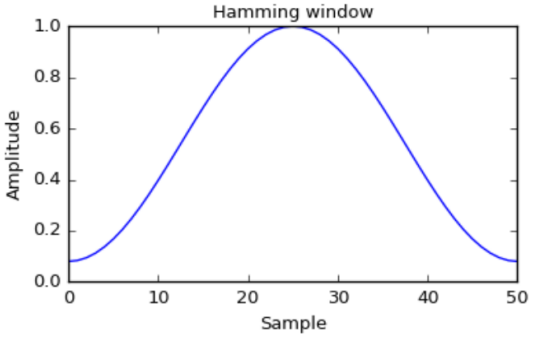
1. Καθαρίζει το directory από μη απαραίτητα αρχεία.
2. Κατεβάζει τα απαραίτητα δεδομένα και τα απαραίτητα scripts.
3. Εκτελεί τις υλοποιήσεις της άσκησης, χρησιμοποιώντας τα παραπάνω scripts και τις αντίστοιχες εντολές.

Μέρος 2: Θεωρητικό υπόβαθρο

Mel-frequency Cepstral Coefficients (MFCCs)

Οι συντελεστές Mel Frequency Cepstral (MFCCs) θεωρούνται μία από τις πιο διαδεδομένες μεθόδους για την εξαγωγή και αναπαράσταση χαρακτηριστικών στην αναγνώριση φωνής. Τα MFCCs βασίζονται στο Cepstrum, το οποίο αποτελεί τον αντίστροφο μετασχηματισμό Fourier του λογαρίθμου του εκτιμώμενου σήματος. Ο μηχανισμός εξαγωγής χαρακτηριστικών περιλαμβάνει τα ακόλουθα βήματα:

- **Pre-Emphasis:** Σε αυτό το στάδιο, γίνεται μια επισήμανση στις υψηλές συχνότητες για να ενισχυθούν. Η ενέργεια που βρίσκεται σε αυτές τις συχνότητες των φωνηέντων είναι συνήθως μικρότερη, και αυτή η τεχνική επιτρέπει καλύτερη ακρίβεια στο μοντέλο.
- **Framing and Windowing:** Η ομιλία επεξεργάζεται σε μικρά χρονικά παράθυρα με τη χρήση παραθύρων, όπως το Hamming, προκειμένου να ληφθούν αξιόπιστα φασματικά χαρακτηριστικά.



Σχήμα 1: Hamming Window

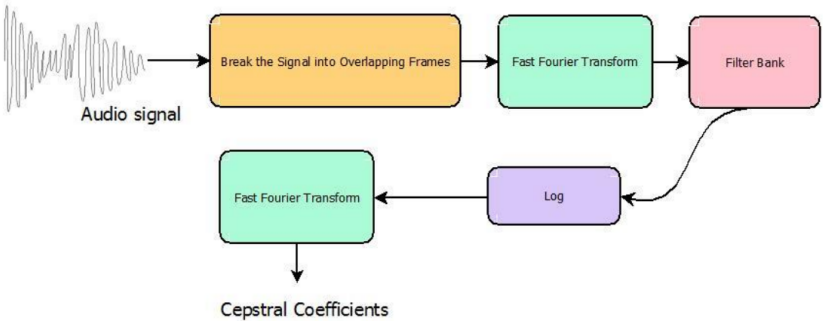
- **Μετασχηματισμός Fourier:** Σε αυτό το στάδιο, εφαρμόζω τον Διακριτό Μετασχηματισμό Fourier (Discrete Fourier Transform - DFT), χρησιμοποιώντας την τεχνική Fast Fourier Transform (FFT) σε N σημεία, για κάθε παράθυρο με μέγεθος N. Αυτό μου επιτρέπει να πάρω τον μετασχηματισμό Short-Time Fourier για το σύνολο του σήματος, διαχωρίζοντας το σε μικρά τμήματα χρόνου και αναλύοντας τη συχνότητά του σε κάθε τμήμα.
- **Mel-Filter-Bank:** Το ανθρώπινο αυτί δεν αντιλαμβάνεται τον ήχο με γραμμικό τρόπο, δηλαδή δεν είναι εξίσου ευαίσθητο σε όλες τις συχνότητες. Συγκεκριμένα, οι αλλαγές στις χαμηλές συχνότητες είναι πιο ευδιάκριτες σε σχέση με αυτές στις υψηλές. Η κλίμακα Mel είναι μια

μη γραμμική κλίμακα που χρησιμοποιείται στην ανάλυση ήχου, λαμβάνοντας υπόψη αυτήν την ιδιότητα της ανθρώπινης ακουστικής αντίληψης.

Η κλίμακα Mel εφαρμόζει ένα μετασχηματισμό στις συχνότητες, καθιστώντας τις πιο συμβατές με την ανθρώπινη αντίληψη. Αυτό σημαίνει ότι οι συχνότητες χαμηλότερων τόνων μετατρέπονται σε αντίστοιχες Mel συχνότητες με περισσότερη ακρίβεια και επισημότητα, ενώ οι υψηλότερες συχνότητες διατηρούνται σε λιγότερο ευαίσθητα επίπεδα.

- **DCT:** Ο Διακριτός Μετασχηματισμός Συνημιτόνου (Discrete Cosine Transform - DCT) είναι ένας μετασχηματισμός που αναπαριστά μια ακολουθία δεδομένων ως γραμμικό συνδυασμό συνημιτόνων διαφορετικών συχνοτήτων. Αντίθετα με τον μετασχηματισμό Fourier, ο DCT δεν περιλαμβάνει φανταστικές συνιστώσες, κάτι που έχει ως αποτέλεσμα γρηγορότερους υπολογισμούς.

Μολονότι ο DCT παράγει μόνο πραγματικές τιμές, έχει αποδειχθεί ότι δεν υπάρχει απώλεια πληροφορίας από το σήμα καθώς είναι γραμμικός και καταφέρνει να αναπαραστήσει τα δεδομένα αποτελεσματικά. Η εφαρμογή του DCT στον λογάριθμο της κλίμακας Mel είναι ιδιαίτερα χρήσιμη στην εξαγωγή χαρακτηριστικών φωνής για την αναγνώριση ομιλίας. Συγκεκριμένα, όταν εφαρμόζεται στον μετασχηματισμό του λογαρίθμου της κλίμακας Mel, ο DCT παράγει 13 συντελεστές cepstral, οι οποίοι αντιπροσωπεύουν τα χαρακτηριστικά του φωνητικού σήματος. Επιπλέον, υπολογίζονται και οι παράγωγοί των αυτών των συντελεστών, γνωστοί ως δέλτα (deltas) και δέλτα-δέλτα (deltas-deltas), που παρέχουν επιπλέον πληροφορίες σχετικά με τη δυναμική του φωνητικού σήματος με την πάροδο του χρόνου.



Σχήμα 2: Enter Caption

Γλωσσικά Μοντέλα (Language Models)

Οι προσεγγίσεις που χρησιμοποιούνται για τη διάκριση μεταξύ ομόηχων λέξεων και προτάσεων βασίζονται σε διάφορες πιθανοτικές μεθόδους και μοντέλα γλώσσας. Ο στόχος είναι να επιλεχθεί ο πιο πιθανός συνδυασμός λέξεων που αποτελεί μια έγκυρη πρόταση στην εκάστοτε γλώσσα. Ορισμένες από τις πιθανοτικές προσεγγίσεις που χρησιμοποιούνται περιλαμβάνουν:

N-gram: Δημιουργεί μια πιθανοτική κατανομή για μια ακολουθία από NN στοιχεία.

$$P(w_1^n) = \prod_{k=1}^{n} P(w_k | w_{k-N+1}^{k-1})$$

Unigram και Bigram: Τα unigram και bigram αποτελούν ειδικές περιπτώσεις του n-gram, με n=1n=1 για το unigram και n=2n=2 για το bigram. Στο bigram, κάθε λέξη εξαρτάται από την προηγώνη κατάσταση, ενώ στο unigram εξαρτάται από την απροτεταρισμένη πιθανότητα εμφάνισής της.

$$\textbf{Unigram:} P(w_n|w_1, w_2, .., w_n) = P(w_n)$$

$$\textbf{Bigram:} P(w_n|w_1, w_2, ..., w_n) = P(w_n|w_{n-1})$$

Φωνητικά Μοντέλα (Acoustic Models)

Τα φωνητικά μοντέλα αντιπροσωπεύουν την αναπαράσταση του φωνητικού σήματος με τη χρήση φωνημάτων. Τα γνωρίσματα του φωνητικού μοντέλου παράγονται με τη χρήση Γκαουσιανών μίξεων. Κάθε πρόταση διασπάται σε επιμέρους λέξεις, οι οποίες στη συνέχεια διασπώνται σε αντίστοιχα φωνήματα. Αυτές οι διαδικασίες έχουν ως στόχο τη δημιουργία αναπαραστάσεων που χρησιμοποιούνται στην εκπαίδευση μοντέλων.

Στη συνέχεια, γίνεται η εκτίμηση των παραμέτρων μεταξύ των καταστάσεων του μοντέλου και δημιουργείται ένας πίνακας μεταβάσεων για κάθε φώνημα. Οι πιθανότητες των παρατηρήσεων αντιπροσωπεύονται από τα Hidden Markov Models (HMMs).

Τέλος, με την υλοποίηση του αλγορίθμου forward-backward γίνεται η εκτίμηση για το ποια είναι η πιθανή ακολουθία φωνημάτων, έτσι ώστε να γίνει η αναγνώριση του φωνητικού σήματος.

Μέρος 3: Βήματα προπαρασκευής

Κατά την προετοιμασία της εργαστηριακής άσκησης, εξοικειωνόμαστε με το Kaldi, κατεβάζω τα απαραίτητα αρχεία χρησιμοποιώντας το get_deps.sh και δημιουργώ τον φάκελο εργασίας (working directory) όπως απαιτείται. Κατόπιν, εκτελώ το part-3.py και:

- Εντός του φακέλου 'egs', δημιουργώ έναν νέο φάκελο με την ονομασία «usc».
- Δημιουργώ τον φάκελο 'data' και τους υποφακέλους «train», «dev», και «test».
- Δημιουργώ σε κάθε φάκελο τα αρχεία:
  - uttids
  - utt2spk
  - wav.scp
  - text
- Χρησιμοποιώντας το αρχείο lexicon.txt, αντικαθιστώ τις λέξεις στις προτάσεις με τις αντίστοιχες ακολουθίες φωνηέντων. Κάνω μια προεπεξεργασία στο κείμενο, μετατρέποντας τους χαρακτήρες σε πεζούς και αφαιρώντας ειδικούς χαρακτήρες. Επιπλέον, προσθέτω το φωνήεν "sil" στην αρχή και στο τέλος κάθε πρότασης.

Μέρος 4: Βήματα κυρίως μέρους

Προετοιμασία διαδικασίας αναγνώρισης φωνής για τη USC-TIMIT

Κατά την εκτέλεση του part4-1.py, ακολουθούνται τα παρακάτω βήματα σύμφωνα με την εκφώνηση:

1. Αντιγράφω τα αρχεία path.sh και cmd.sh από τον φάκελο wsj. Στο αρχείο path.sh, ορίζω σωστά τη μεταβλητή KALDI_ROOT και στο αρχείο cmd.sh θέτω τις μεταβλητές train_cmd, decode_cmd και cuda_cmd ώστε να χρησιμοποιούν το run.pl.
2. Δημιουργώ soft links με τα ονόματα steps και utils μέσα στον φάκελο usc, οι οποίοι θα δείχνουν στους αντίστοιχους φακέλους του wsj.
3. Δημιουργώ τον φάκελο conf και εκτελώ αντιγραφή του αρχείου mfcc.conf από τη διαδρομή slp-ntua/slp-labs σε αυτόν.
4. Δημιουργώ τους φακέλους: data/lang, data/local/dict, data/local/lm_tmp, data/local/nist_lm

Προετοιμασία γλωσσικού μοντέλου

Τρέχοντας το part4-2.py εκτελούνται τα παρακάτω βήματα:

1. Αρχικά, στον φάκελο 'data/local/dict' αποθηκεύονται τα αρχεία που απαιτούνται για την κατασκευή του γλωσσικού μοντέλου.
   (α′) Τα αρχεία silence_phones.txt και optional_silence.txt περιέχουν μόνο το φώνημα της σιωπής (sil).
   (β′) Το αρχείο 'nonsilence_phones.txt' περιέχει τα υπόλοιπα φωνήματα ταξινομημένα.
   (γ′) Το αρχείο lexicon.txt αντιστοιχίζει κάθε φωνήεν στον εαυτό του.
   (δ′) Τα αρχεία 'lm_train.text', 'lm_dev.text', και 'lm_test.text' προκύπτουν από τα αντίστοιχα αρχεία 'text' με την προσθήκη των '<s>' και '</s>' σε κάθε πρόταση στην αρχή και στο τέλος αντίστοιχα.
   (ε′) Το κενό αρχείο extra_questions
2. Στον φάκελο 'data/local/lm_tmp', δημιουργώ την ενδιάμεση μορφή του γλωσσικού μοντέλου χρησιμοποιώντας το πακέτο IRSTLM και την εντολή 'build-lm.sh'. Πριν εκτελέσω αυτή την εντολή, ενεργοποιώ το 'path.sh' ώστε το αντίστοιχο script να είναι διαθέσιμο. Με την 'build-lm.sh', δημιουργώ unigram και bigram μοντέλα, θέτοντας την παράμετρο 'n' ίση με 1 για το unigram και ίση με 2 για το bigram.
3. Στον φάκελο 'data/local/nist_lm', αποθηκεύω το compiled γλωσσικό μοντέλο σε μορφή ARPA χρησιμοποιώντας την εντολή 'compile-lm.sh'. Πριν εκτελέσω αυτή την εντολή, ενεργοποιώ το 'path.sh' ξανά για να είναι διαθέσιμο το αντίστοιχο script.

   Αφού εκτελέσω το 'compile-lm.sh', προκύπτουν τα αρχεία 'lm_phone_ug.arpa.gz' για το unigram μοντέλο και 'lm_phone_bg.arpa.gz' για το bigram μοντέλο.
4. Στον φάκελο data/lang, δημιουργώ το L.fst, που είναι το FST (Finite State Transducer) του λεξικού της γλώσσας, χρησιμοποιώντας την εντολή prepare_lang.sh, και έπειτα την καλώ.
5. Τα αρχεία wav.scp, text, και utt2spk ταξινομούνται με τη χρήση της εντολής sort στους φακέλους data/train, data/dev, και data/test.

6. Χρησιμοποιώ το script 'utils/utt2spk_to_spk2utt.pl' για να δημιουργήσω το αρχείο 'spk2utt'.
7. Τέλος, δημιουργώ το 'G.fst' (FST γραμματικής) χρησιμοποιώντας τη διαδικασία TIMIT, όπως περιγράφεται στο αρχείο 'local/timit_format_data.sh'.

Ερώτημα 1: Για τα γλωσσικά μοντέλα που δημιουργήσατε υπολογίστε το perplexity στο validation και στο test set. Τι δείχνουν αυτές οι τιμές;

Το Perplexity είναι ένα μέτρο που χρησιμοποιείται συχνά στα πιθανοτικά μοντέλα γλωσσικής αναγνώρισης και μετάφρασης για να αξιολογήσει την απόδοση του μοντέλου. Υπολογίζει πόσο καλά μπορεί ένα πιθανοτικό μοντέλο να προβλέψει ένα δείγμα από ένα σύνολο δεδομένων.

Το Perplexity αξιολογείται καλύτερα όταν είναι όσο το δυνατόν χαμηλότερο. Ένα χαμηλό Perplexity υποδεικνύει ότι το πιθανοτικό μοντέλο είναι πιο "σίγουρο" ή πιο ακριβές στις προβλέψεις του, ενώ ένα υψηλό Perplexity υποδεικνύει μεγαλύτερη αβεβαιότητα ή περιπλοκότητα στο μοντέλο.

Το Perplexity είναι ιδιαίτερα χρήσιμο για να συγκρίνω διαφορετικά πιθανοτικά μοντέλα ή να παρακολουθώ την πρόοδο της εκπαίδευσης ενός μοντέλου κατά τη διάρκεια της εκπαίδευσης.

	Unigram	Bigram
Dev	56.23	27.02
Test	55.15	26.39

Παρατηρώ ότι το bigram αποδίδει πολύ καλύτερα από το unigram, καθώς το Perplexity πέφτει σχεδόν στο μισό. Αυτό είναι λογικό αποτέλεσμα, διότι το unigram δεν λαμβάνει υπόψη του την προηγούμενη κατάσταση για να υπολογίσει την πιθανότητα εμφάνισης κάθε φωνήεντος, αλλά μόνο την a-priori πιθανότητα εμφάνισης του.

Στο unigram μοντέλο, κάθε φωνήεν έχει μια ανεξάρτητη πιθανότητα εμφάνισης, χωρίς να λαμβάνεται υπόψη η σειρά των φωνηέντων. Από την άλλη πλευρά, το bigram μοντέλο λαμβάνει υπόψη το προηγούμενο φωνήεν και υπολογίζει την πιθανότητα εμφάνισης ενός φωνήεντος βάσει του προηγούμενου φωνήεντος.

Έτσι, το bigram μοντέλο μπορεί να αποτυπώσει καλύτερα τις σχέσεις μεταξύ των φωνηέντων σε μια ακολουθία, ενώ το unigram μοντέλο απλώς υπολογίζει τις ανεξάρτητες πιθανότητες εμφάνισης των φωνηέντων.

Συνεπώς, η καλύτερη απόδοση του bigram σε σχέση με το unigram στη μείωση του Perplexity αντικατοπτρίζει την ικανότητά του να αντιλαμβάνεται καλύτερα τη συνεκτικότητα και τη σειρά των φωνηέντων στη γλώσσα.

Εξαγωγή ακουστικών χαρακτηριστικών

Κατά την εκτέλεση του 'part4-3.py', το οποίο εκτελεί τα scripts 'compute_cmvn_stats.sh', εξάγω τα MFCCs για τα τρία σύνολα δεδομένων.

Ερώτημα 2: Με τη δεύτερη εντολή πραγματοποιείται το λεγόμενο Cepstral Mean and Variance Normalization. Τι σκοπό εξυπηρετεί;

Το CMVN (Cepstral Mean and Variance Normalization) είναι μια αποδοτική τεχνική κανονικοποίησης σήματος που χρησιμοποιείται στην αναγνώριση φωνής. Ο στόχος της είναι να μειωθεί η παραμόρφωση που προκαλείται από το θόρυβο στα φωνητικά σήματα, προκειμένου να βελτιστοποιηθεί η εξαγωγή χαρακτηριστικών. Αυτό επιτυγχάνεται με την κανονικοποίηση της μέσης τιμής και της διασποράς των σημάτων.

Η κανονικοποίηση αυτή στοχεύει στην ομαλοποίηση της διασποράς των σημάτων. Καθώς διαμορφώνονται οι συντελεστές των φωνητικών χαρακτηριστικών, μικρές αλλαγές και θόρυβος μπορούν να προκαλέσουν μεγάλη διακύμανση (variance), με αποτέλεσμα να δυσκολεύεται η σωστή αναγνώριση των φωνημάτων.

Η CMVN επιτυγχάνει τη σταθεροποίηση της διακύμανσης και εξομαλύνει τις διαφορές που προκαλεί ο θόρυβος, βοηθώντας έτσι στην ακριβή αναγνώριση των φωνημάτων από το σύστημα αναγνώρισης φωνής.

Ερώτημα 3: Πόσα ακουστικά frames εξήχθησαν για κάθε μία από τις 5 πρώτες προτάσεις του training set; Τι διάσταση έχουν τα χαρακτηριστικά;

Βασιζόμενη στο αρχείο data/train/utt2num_frames, έχω τις παρακάτω πληροφορίες για τη διάρκεια (σε frames) κάθε ηχητικού αρχείου:

- f1_003: 317 frames
- f1_004: 371 frames
- f1_005: 399 frames
- f1_007: 328 frames
- f1_008: 464 frames

Η διάσταση των χαρακτηριστικών υπολογίζεται ως εξής:

$$\text{Διάσταση} = 13 \times \left(\frac{\text{διάρκεια}}{\text{frame shift}} \right)$$

Όπου:

- Διάσταση: Η διάσταση των χαρακτηριστικών.
- 13: Ο αριθμός των συντελεστών του Mel-Frequency Cepstrum.
- διάρκεια: Η διάρκεια του ηχητικού αρχείου σε δευτερόλεπτα.
- frame shift: Η διάρκεια του frame shift σε δευτερόλεπτα.

Εκπαίδευση ακουστικών μοντέλων και αποκωδικοποίηση προτάσεων

Τρέχοντας το part4-4.py εκτελούνται τα παρακάτω βήματα:

1. Χρησιμοποιώντας το script 'steps/train_mono.sh', εκπαιδεύω ένα monophone GMM-HMM ακουστικό μοντέλο πάνω στο σύνολο εκπαίδευσης (train dataset).
2. Για τη δημιουργία του γράφου HCLG σύμφωνα με τη γραμματική G που προέκυψε από το προηγούμενο βήμα, χρησιμοποιώ το script 'utils/mkgraph.sh'.
3. Για την αποκωδικοποίηση των προτάσεων στα δεδομένα επικύρωσης (validation) και στα δεδομένα ελέγχου (test) χρησιμοποιώντας τον αλγόριθμο Viterbi, χρησιμοποιώ το script 'steps/decode.sh'.
4. Χρησιμοποιώ το local/score.sh και τα αποτελέσματα που βρίσκονται στα αρχεία:
   (α′) exp/mono/decode_dev_ug/scoring_kaldi/best_wer
   (β′) exp/mono/decode_dev_bg/scoring_kaldi/best_wer
   (γ′) exp/mono/decode_test_ug/scoring_kaldi/best_wer
   (δ′) exp/mono/decode_test_bg/scoring_kaldi/best_wer

   Η μετρική PER (Phone Error Rate) υπολογίζεται με τον ακόλουθο τρόπο:

$$\text{PER } (\%) = 100 \times \frac{\text{insertions} + \text{substitutions} + \text{deletions}}{\text{phonemes}}$$

   Όπου:

   - insertions: Ο αριθμός των εισαγωγών (insertions), δηλαδή οι φωνήματα που εισήχθησαν παραπάνω από το πραγματικό.
   - substitutions: Ο αριθμός των αντικαταστάσεων (substitutions), δηλαδή οι φωνήματα που αντικαταστάθηκαν με άλλα λανθασμένα φωνήματα.
   - deletions: Ο αριθμός των διαγραφών (deletions), δηλαδή τα φωνήματα που δεν αναγνωρίστηκαν και έπρεπε να υπάρχουν.
   - phonemes: Ο συνολικός αριθμός των φωνημάτων (phonemes) που περιλαμβάνονται στο σύστημα αναγνώρισης.

   Η μετρική PER εκφράζεται συνήθως ως ποσοστό και χρησιμοποιείται για να αξιολογήσει την απόδοση ενός συστήματος αναγνώρισης φωνής, με χαμηλότερη τιμή PER να υποδεικνύει καλύτερη ακρίβεια και απόδοση στην αναγνώριση. Προκύπτουν τα ακόλουθα αποτελέσματα:

	Unigram	Bigram
Dev	52.66%	52.66%
Test	51.59%	45.01%

5. Κάνω align τα φωνήματα τρέχοντας το steps/align_si.sh, και με βάση τα alignments εκπαιδεύω ένα triphone μοντέλο χρησιμοποιώντας το steps/train_deltas.sh. Δημιουργώ τον γράφο HCLG και πραγματοποιώ αποκωδικοποίηση ξανά με τα αποτελέσματα να βρίσκονται στα αρχεία:
   `exp/tri1/decode_dev_ug/scoring_kaldi/best_wer`
   `exp/tri1/decode_dev_bg/scoring_kaldi/best_wer`
   `exp/tri1/decode_test_ug/scoring_kaldi/best_wer`
   `exp/tri1/decode_test_bg/scoring_kaldi/best_wer`

   και προκύπτει:

	Unigram	Bigram
Dev	40.00%	36.66%
Test	39.07%	35.03%

Παρατηρείται σημαντική βελτίωση έναντι του monophone μοντέλου, αλλά και διαφορές μεταξύ των unigram και bigram.

Ερώτημα 4: Εξηγήστε τη δομή ενός ακουστικού μοντέλου GMM-HMM. Τι σκοπό εξυπηρετούν τα μαρκοβιανά μοντέλα στη συγκεκριμένη περίπτωση και τι τα μίγματα γκαουσιανών; Με ποιο τρόπο γίνεται η εκπαίδευση ενός τέτοιου μοντέλου; Περιγράψτε τη διαδικασία εκπαίδευσης ενός μονοφωνικού μοντέλου.

Τα GMM αποτελούν έναν καλό τρόπο να προσδιορίζονται και να ομαδοποιούνται οι καταστάσεις του αυτόματου σε κατηγορίες, βασιζόμενοι στην πιθανότητα ενός φωνήματος να εμφανίζεται σε μία λέξη. Αν το μοντέλο μου χρησιμοποιούσε αποκλειστικά Gaussian-Mixture-Model, αυτό θα σήμαινε ότι θα κατηγοριοποιούσε κάθε φωνήμα χωρίς να λαμβάνει υπόψη τις προηγούμενες καταστάσεις. Έτσι, δε θα μπορούσε να εξετάσει αν υπάρχει κάποια συνάφεια μεταξύ μιας πρόβλεψης και της συγκεκριμένης λέξης ή πρότασης, με στόχο να ελαχιστοποιήσει την πιθανότητα λάθους.

Γι' αυτόν τον λόγο, αξιοποιούνται τα HMMs (Κρυφά Μαρκοβιανά Μοντέλα), στατιστικά μοντέλα που υποθέτουν την ύπαρξη κρυφών καταστάσεων. Αυτές οι κρυφές καταστάσεις μπορεί να αναπαριστούν την τοποθέτηση γλώσσας, τη συντακτική θέση μιας λέξης στην οποία βρίσκεται το φώνημα, καθώς και τις πιθανότητες μετάβασης από μια ακολουθία N φωνημάτων σε ένα άλλο.

Ο τρόπος εκπαίδευσης χρησιμοποιεί μια ειδική κατηγορία του αλγορίθμου Expectation-Maximization, γνωστή ως forward-backward. Αυτός ο αλγόριθμος μου επιτρέπει να υπολογίσω την πλήρη υπό συνθήκη πιθανοφάνεια μιας σειράς καταστάσεων, δεδομένης μιας ακολουθίας παρατηρήσεων.

Η διαδικασία ξεκινά με μια αρχική εκτίμηση της μέσης τιμής και της διασποράς των δεδομένων. Στη συνέχεια, ο αλγόριθμος επαναλαμβάνει αυτήν τη διαδικασία υπολογισμού με βάση τις προηγούμενες εκτιμήσεις και τις καταστάσεις μετάβασης. Μέσω αυτών των επαναλήψεων, η πιθανοφάνεια αυξάνεται και συγκλίνει σε μια ολοκληρωμένη τιμή.

Με αυτόν τον τρόπο, εκπαιδεύω ταυτόχρονα τόσο τις πιθανότητες μετάβασης (transition probabilities) όσο και τις πιθανότητες εκπομπής (emission probabilities) του HMM. Αυτή η διαδικασία επιτρέπει τη βελτίωση του μοντέλου μου και την προσαρμογή του στις παρατηρήσεις που λαμβάνω, βελτιώνοντας έτσι την ακρίβεια και την απόδοσή του στην ανάλυση των δεδομένων.

Ερώτημα 5: Γράψτε πώς υπολογίζεται η a posteriori πιθανότητα σύμφωνα με τον τύπο του Bayes για το πρόβλημα της αναγνώρισης φωνής. Συγκεκριμένα, πώς βρίσκεται η πιο πιθανή λέξη (ή φώνημα στην περίπτωσή μας) δεδομένης μίας ακολουθίας ακουστικών χαρακτηριστικών;

Το Μπεϋζιανό μοντέλο για την εκτίμηση της πιθανότητας ενός φωνήματος βασίζεται στην ανάλυση a posteriori για κάθε δυνητικό φωνήμα, λαμβάνοντας υπόψη μια ακολουθία λέξεων και φωνημάτων

(χαρακτηριστικών).

$$P(W|X) = \frac{P(X|W)P(W)}{P(X)}$$

όπου:

- $P(W|X)$ η a-posteriori πιθανότητα
- $P(X|W)$ η πιθανότητα ανίχνευσης χαρακτηριστικών δεδομένου του φωνήματοςW i επομένως το φωνητικό μοντέλο(acoustic model)
- $P(W)$ η a-priori που είναι η πιθανότητα εμφάνισης των χαρακτηριστικών(language model)

Αφού διερευνήσω τα i πιθανά φωνήματα που προέκυψαν, εντοπίζω τη μέγιστη πιθανότητα και προκύπτει το εκτιμώμενο φώνημα.

$$W = \arg\max_{0 \le i \le n} \{P(W_i \mid X)\} = \arg\max_{0 \le i \le n} \{P(X \mid W_i) \cdot P(W_i)/P(X)\}$$

Αν θέλω να διαπιστώσω για ποιο i ο όρος $P(X)$ παραμένει ίδιος για κάθε i, τότε έχω το ακόλουθο:

$$W = \arg\max_{0 \le i \le n} \{P(X \mid W_i) \cdot P(W_i)\}$$

Ερώτημα 6: Εξηγήστε τη δομή του γράφου HCLG του Kaldi περιγραφικά.

Ο γράφος HCLG είναι ένα αναπτυγμένο γράφημα αποκωδικοποίησης που αναπαριστά τον αποδοχέα γραμματικής, το λεξικό φωνημάτων (lexicon), τα HMMs και τις συναρτησιακές εξαρτήσεις (context-dependency). Το αποτέλεσμα είναι ένα προσανατολισμένο στο μέλλον (fst) με αναγνωριστικά λέξεων (word-ids), και ως είσοδο δέχεται αναγνωριστικά λέξεων (word-ids).

- Το FST H περιέχει πληροφορίες για τα Hidden Markov Models (HMMs).
- Το FST C εφαρμόζει το context-dependency στα φωνήματα.
- Το FST L αντιπροσωπεύει το λεξικό, δέχοντας ως είσοδο μία λέξη και επιστρέφοντας τα επιμέρους φωνήματά της.
- Το FST G είναι ο αποδοχέας της γραμματικής (grammar acceptor).