Αναφορά 2ου εργαστηρίου Επεξεργασίας Φωνής

Θύμιος Τζίνης 03112007
Ευάγγελος Χατζηπανταζής 03112050

Θεωρητική Ανάλυση Εννοιών.

Γλωσσικό Μοντέλο (Language Model LM):

Για το μοντέλο αυτό συνήθως χρησιμοποιούμε 4-grams ή tri-grams και πιό σπάνια unigrams και bigrams. Ωστόσο, κατά την χρησιμοποίηση του Kaldi για την δημιουργία του μοντέλου αναγνώρισης φωνής βλέπουμε ότι δηιουργεί unigram, bigram και trigram συνδυάζοντάς τα και προφανώς αυξάνοντας την υπολογιστική πολυπλοκότητα αλλά και την ακρίβεια όσο αυξάνεται το μέγεθος του μοντέλου.

Ακουστικό Μοντέλο (Acoustic Model AM):

Στο στάδιο της δημιουργίας του ακουστικού μοντέλου ουσιαστικά υπολογίζουμε την πιο πιθανή ακολουθία από παρατηρήσεις όταν μας έχουν δοθεί άλλα γλωσσικά χαρακτηριστικά όπως (λέξεις, φωνήματα ή κάποια μέρη φωνημάτων). Πιο συγκεκριμένα δοθέντος μίας πιθανότητας P(O|W) μπορούμε να κατατάξουμε για κάθε HMM μίας κατάστασης q του αυτομάτου μας, χρησιμοποιώντας Gaussian Mixture Models (GMM's) και τα υπόλοιπα γλωσσικά χαρακτηριστικά, την πιθανοφάνεια.

Συνολικό Μοντέλο Αναγνώρισης φωνής:

Ένα απλό θεωρητικό μοντέλο για την κάθε λέξη θα ήταν το:

$$\hat{W} = \underset{W \in \mathcal{L}}{\operatorname{argmax}} P(O|W) P(W)^{LMSF}$$

το οποίο βλέπουμε ότι περιέχει και ένα LMSF language model scaling factor που ουσιαστικά προδίδει την εξάρτηση του μοτέλου μας από το γλωσσικό μοντέλο και πρακτικά καταφέρνει να ελέγξει το ποσοστό επίδρασής του(όσο αυξάνουμε αυτό το βάρος τόσο λιγότερο μετράει στο μοντέλο μας η γλωσσική επιρροή).

Ένα πιο πρακτικό μοντέλο που χρησιμοποιούμε είναι το:

$$\hat{W} = \underset{W \in \mathcal{L}}{\operatorname{argmax}} \log P(O|W) + LMSF \times \log P(W) + N \times \log WIP$$

Το P(O|W) αφορά το ακουστικό μας μοντέλο (observation likelihood).
Το P(W) αφορά το γλωσσικό μας μοντέλο (prior probability).
Που χρησιμοποιεί λογαριθμικές πιθανότητες κόστους.

Κάθε φορά που εισάγουμε μία λέξη στο μοντέλο μας πληρώνουμε ένα αντίστοιχο τίμημα που ονομάζεται WIP word insertion penalty. Στο μοντέλο μας χρησιμοποιούμε στην φάση της αποκωδικοποίησης τον αλγόριθμο Viterbi για μεγαλύτερη ταχύτητα συνδυάζοντας το γλωσσικό (για το Kaldi έχουμε tri-gram grammar) και ακουστικό μας μοντέλο (acoustic likelihoods) για να βγάλουμε την πιο πιθανή πρόταση από λέξεις.

Λίγα Λόγια για MFCC:

Η διαδικασία της εκπαίδευσης ενός συστήματος αναγνώρισης φωνης,περνάει μέσα από την εξαγωγή δεδομένων χαρακτηριστικών.Η πιο διαδεδομένη μορφή απεικόνισης αυτών είναι ένα MFCC Vector 39 θέσεων.Η ανάπτυξη της ιδέας αυτής χρησιμοποιεί και ιδέες από cepstrum,καθώς και ψυχοακουστικές μελέτες για την μετάφραση του ήχου στην κατάλληλη μορφή .
Η εξαγωγή των χαρακτηριστικών τμηματοποιείται σε 6 στάδια:
1)**Προέμφαση του ήχου** μέσα από την μετατόπιση κάθε φωνήματος σσε υψηλότερες συχνότητες που βοηθούν την αναγνώριση.
2)**Παραθυροποίηση των ηχητικών δεδομένων** για να διατηρήσουμε την στατικότητα των στατιστικών χαρακτηριστικών του ήχου(μέση τιμή,διακύμανση) και κατεπέκταση την εργοδικότητα(αφού έχουμε πλήθος δεδομένων στο χρόνο αλλά δε μπορούμε να επαναλάβουμε διαδοχικά τις ίδιες εκτελέσεις των ίδιων δεδομένων).Συχνά χρησιμοποιούμε Kaiser παράθυρα με δεδομένο overlap για να ομαλοποιηθούν φαινόμενα των άκρων.
3)**Συχνοτική ανάλυση σήματος** ,μέσω τηης χρήσης DFT για τον υπολογισμό της συγκέντρωσης της ενέργειας σε κάθε συχνότητα.
4)**Mel filter bank διαμόρφωση,**που αιτιολογείται επειδή ο ανθρώπινος εγκέφαλος παρουσιάζει δυσκολία στον διαχωρισμό συχνοτήτων σε υψηλοσυχνοτικά σήματα.Γενικά η ανθρώπινη απόκριση είναι γραμμική κάτω από τα 1000Hz και λογαριθμική παραπάνω,οπότε έτσι σχεδιάονται και τα φίλτρα μας.
5)**Χρήση cepstrum,**για την μετάθεση του σήματος σε πεδίο quefrency και το διαχωρισμό πηγής και φίλτρου.Αποδεικνύεται ότι ο διαχωρισμός αυτός βελτιώνει το μονελο καθώς ο εντοπισμός π.χ. του φίλτρου: Θέση της γλώσσας μπορεί να προσφέρει στην ανάλυση παραπάνω χαρακτηριστικά για την αναγνώριση των φωνημάτων.
6)**Delta-Training**: Τα χαρακτηριστικά αυτά είναι τιμές που δείχνουν πόσο μεταβάλλονται κάποια στατιστικά χαρακτηριστικά του cepstrum μέσα σε ένα παράθυρο.Συμπληρώνουν το διάνυσμα που δημιουργούμε.Για ακρίβεια χρησιμοποιούμε και τα double-deltas που δείχνουν την μεταβολή των deltas .

Αυτός ο χειρισμός του σήματος καταλήγει σε 39 MFC Coeficients και περιέχει το μεγαλύτερο μέρος της πληροφορίας για κάθε παραθυροποιημένο σήμα.

Ανάλυση βημάτων κώδικα:

Το αρχείο που περιέχει κάθε ένα από τα βήματα της εκφώνησης είναι το ***vag_thy_run.sh*** που συμπεριλαμβάνουμε στον φάκελο.Το αρχείο αυτό βασίστηκε σε μεγάλο βαθμό στο **run.sh** ως προς την ακολουθία τον βημάτων.Παρακάτω εξηγούνται αυτά τα βήματα καθώς και οι αλλαγές που έγιναν.
Καταρχάς,το ***vag_thy_run.sh*** τοποθετείται στον φάκελο εκκίνησης που είναι ο **home/<onoma_xrhsth>/kaldi-trunk/egs/wsj/s5.** Στον ίδιο φάκελο τοποθετούμε και τον **WSJ02015** φάκελο που περιέχει τα φωνητικά δεδομένα(Wall Street Journal).

Επιπλέον,στο αρχείο **cmd.sh ,**όπως αναφέρεται στις οδηγίες αυτού,επειδή δεν χρησιμοποιούμε το GridEngine,τρέχουμε τον εξής κώδικα:
#b) run it locally...
export train_cmd=run.pl
export decode_cmd=run.pl
export cuda_cmd=run.pl
export mkgraph_cmd=run.pl

(Το αρχείο περιλαμανεται στην αναφορά).

Βήμα1:
Αρχικά πρέπει να κάνουμε μια προεργασία στα δεδομένα των συνεντεύξεων που δίνονται για να ανγνωρίζονται από το Kaldi.Τα δεδομένα αυτά περιέχονται στον φάκελο **WSJ0/11 13.1/wsj0/doc/indices/train/.**Επειδή λείπουν τα wsj1 δεδομένα έγιναν κάποιες αλλαγές στο αρχείο **wsj_data_prep.sh**,και για αυτόν τον λόγο το συμπεριλαμανουμε στην αναφορά.Η διαδικασία του data preparation τερματίζεται με το format των δεδομένων στην κατάλληλη μορφή.
Τα μηνύματα Data Preparation succeeded και Succeeded in formatting data,σηματοδοτούν το τέλος του βήματος 1.

Ενδιάμεσα αποτελέσματα:

1)Διάβασμα αρχείων (συνεχίζει και γιαάλλα αρχεία)***:***

loadtxt_ram()
1-grams: reading 4989 entries
done level 1
2-grams: reading 1639687 entries
done level 2
3-grams: reading 2684151 entries
done level 3
done
starting to use OOV words [<unk>]
OOV code is 4989
OOV code is 4989
OOV code is 4989
pruning LM with thresholds:
1e-07 1e-07
ng: <s> 0 nextlevel_ts=1.99968 nextlevel_tbs=0.931817 k=1 ns=4206

2) Data preparation succeeded
3) local/wsj_prepare_dict.sh --dict-suffix _nosp
Checked out revision 13156.
Dictionary preparation succeeded
4) --> data/local/dict_nosp/silence_phones.txt is OK
--> data/local/dict_nosp/optional_silence.txt is OK
--> data/local/dict_nosp/nonsilence_phones.txt is OK
--> disjoint property is OK. etc.
5) --> SUCCESS [validating lang directory data/lang_nosp]
6) Preparing language models for test
arpa2fst -
Processing 1-grams
Processing 2-grams
Connected 0 states without outgoing arcs.
remove_oovs.pl: removed 53 lines.

(Αυτό συνεχίζεται και για άλλα αρχεία).

7) Succeeded in formatting data.

Βήμα2:

Σε αυτό το βήμα εξάγονται όλα τα χαρακτηριστικά των φωνητικών δεδομένων σε μορφή ενός vector,που αποτελείται απο 39 MFC Coefficients,που θα αποτελέσουν την βάση για την κατασκευή του συστήματων αναγνώρισης φωνής και την δημιουργία δημιουργία γλωσσικών μοντέλων και ακουστικών μοντέλων αναγνώρισης φωνημάτων.Στην διαδικασία αυτή εκτελούνται τα περισσότερα σημεία από αυτά που περιγράφηκαν παραπάνω:Μετατόπιση των ηχητικών δεδομένων

στις κατάλληλες συχνότητες,Κατάτμηση των δεδομένων σε μικρά παράθυρα για υπολογισμό MFCC σε κάθε ένα,Χρήση filterbank που προέρχεται από ψυχοακουστικές μελέτες,delta-training,υπολογισμός Ενεργειών φιλτραρισμένων σημάτων,Επαναφορά του σήματος στον χρόνο για εξαγωγή χχαρακτηριστικών φίλτων και πηγής.Στο τελευταίο βήμα έγινε χρήση του iDCT αντί για iDFT καθώς ο μετασχηματισμός ημιτόνου συγκεντρώνει καλύτερα τις ενέργειες γύρω από τις ζητούμενες χαμηλές συχνότητες.

Κάθε ένα από τα παραπάνω σημεία εκτελείται από ένα script ενώ τα τελικά αποτελέσματα περιέχονται στον φάκελο mfcc.Προφανώς έγιναν οι απαραίτητες αλλαγές για να περιλαμβάνονται μόνο τα αρχεία που έχουμε στην διάθεσή μας **(train_si84, test_eval92, test_eval92_5k)**,όπως φαίνεται στο **vag_thy_run.sh**.

Ενδιάμεσα αποτελέσματα:

1) **utils/validate_data_dir.sh: Successfully validated data-directory data/train_si84**

(Συνεχίζει και για τα άλλα αρχεία).

2) **Succeeded creating MFCC features for test_eval92**

Succeeded creating CMVN stats for test_eval92_5k

Στην συνέχεια για την τμηματοποίηση χρησιμοποιήθηκε όλο το αρχείο δεδομένων αφού παρατηρήσαμε καλύτερα αποτελέσματα στ WER αρχεία.

Υπόλοιπα Βήματα:

Από εδώ και πέρα ξεκινάει η εκπαίδευση του μοντέλου για το χτίσιμο του αναγνωριστή και την εξαγωγή των τελικών αποτελεσμάτων.

Τελικά, αφού έχουμε ολοκληρώσει το μοντέλο μας ήρθε η ώρα να το τεστάρουμε σε κάποια test data που μας δίνονται. Το μοντέλο μας στα νέα πλέον δεδομένα προσπαθεί να κάνει την κατάλληλη αποκωδικοποίηση και τα αποτελέσματα φαίνονται στα αρχεία wer (Word Error Ratio) που βρίσκονται στο path: **/kaldi-trunk/egs/wsj/s5/exp/tri1/decode_nosp_tgpr_eval92** και τώρα τρέχουμε το κατάλληλο bash file που βρίσκεται στο path: **utils/best_wer.sh** προκειμένου να τυπώσουμε στο αρχείο μας:
/kaldi-trunk/egs/wsj/s5/exp/tri1/decode_nosp_tgpr_eval92/scoring_kaldi/best_wer
το καλύτερο scoring σε WER το οποίο τυπώνεται και στην οθόνη μαζί με τα insertions, deletions, substitutions που κάνει κάθε φορά το Kaldi για κάθε data test.

Παρατίθεται το τελικό αποτέλεσμα από το running του: **bash ./vag_thy_run.sh**

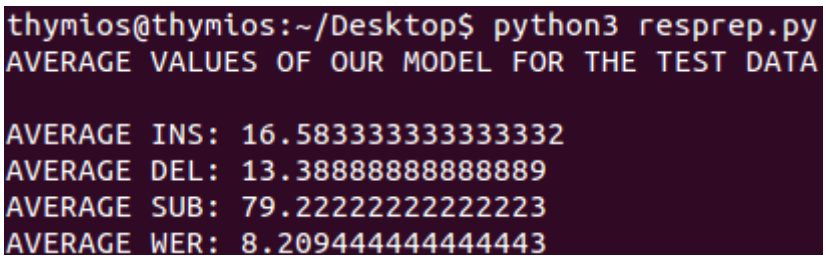
%WER 7.37 [98 / 1330, 11 ins, 14 del, 73 sub] exp/tri1/decode_nosp_tgpr_eval92/wer_17_0.5

Μπορούμε εύκολα να δούμε και τα άλλα scores από όλα τα άλλα αρχεία wer στον αντίστοιχο φάκελο **decode_nosp_tgpr_eval92.**

Εκτελούμε στο bash:

```
thymios@thymios:~/kaldi-trunk/egs/wsj/s5/exp/tri1/decode_nosp_tgpr_eval92$ echo wer
wer_10_0.0 wer_10_0.5 wer_10_1.0 wer_11_0.0 wer_11_0.5 wer_11_1.0 wer_12_0.0 wer_12
0.5 wer_12_1.0 wer_13_0.0 wer_13_0.5 wer_13_1.0 wer_14_0.0 wer_14_0.5 wer_14_1.0 we
_15_0.0 wer_15_0.5 wer_15_1.0 wer_16_0.0 wer_16_0.5 wer_16_1.0 wer_17_0.0 wer_17_0.
 wer_17_1.0 wer_18_0.0 wer_18_0.5 wer_18_1.0 wer_19_0.0 wer_19_0.5 wer_19_1.0 wer_2
_0.0 wer_20_0.5 wer_20_1.0 wer_9_0.0 wer_9_0.5 wer_9_1.0
thymios@thymios:~/kaldi-trunk/egs/wsj/s5/exp/tri1/decode_nosp_tgpr_eval92$ cat wer*
> ~/Desktop/results.txt
```

Και δημιουργούμε στο Desktop ένα αρχείο results.txt που έχει όλα αυτά τα αποτελέσματα. Γράψαμε ένα script σε python προκειμένου να εξάγουμε κάποια χαρακτηριστικά για τα αποτελέσματά μας (Το αρχείο μας ονομάζεται **resprep.py**) κυρίως μέσες τιμές. Λαμβάνουμε υπόψιν μας όλα τα αποτελέσματά από τα wer files και έτσι έχουμε το εξής στατιστικό:



Βλέπουμε ότι το μοντέλο μας έχει πάει αρκετά καλά αν αναλογιστούμε τα λίγα δεδομένα που είχαμε στο training και σε όλα τα data test files έχουμε ποσοστό επιτυχίας περίπου 92%! Άρα το μοντέλο μας είναι αρκετά ικανοποιητικό και στο συγκεκριμένο data set με test files είχε τα παραπάνω στατιστικά στοιχεία. Τα αρχεία **results.txt** και **resprep.py** βρίσκονται μέσα στο παραδοτέο φάκελο για εξέταση ή επαλήθευση.

Μπορει κανείς να κοιτάξει και τα log αρχεία που περιλάβαμε στην αναφορά για να δει πως αποτυπώνονται σε μορφή κειμένου τα ηχητικά αποτελέσματα.

Βελτιώσεις Μοντέλου:

Σαν βελτίωση της αλγοριθμικής διαδικασίας μπορούμε να επισημάνουμε το εξής:

1)**Όσον αφορά στην εξαγωγή MFCC:**Έχει υποτεθεί στην ανάλυση η ανεξαρτησία των παραθύρων των σημάτων πέρα της επικάλυψης και η ανάλυση καθενός ως έχει.Αυτό είναι κάτι που ισχύει σπάνια στην διαδοχή των φωνημάτων.Θα προτείναμε όπως και στα γλωσσικά μοντέλα την εισαγωγή Hidden Markov Models και στην εξαγωγή των χαρακτηριστικών για την δημιουργία κάποιας τύπου “συντακτικής” ανάλυσης του λόγου ή έστω κάποιας συσχέτισης στην ανάλυση διαδοχικών φωνημάτων με χρήση bigrams.
2)**Όσον αφορά στην εκπαίδευση του μοντέλου:**Σε γενικές γραμμές,οι ανεξαρτησίες φωνημάτων είναι σπάνιες στην διατύπωση μιας πρότασης.Αυτός είναι και ο λόγος που προσπαθούμε να συγκεντρώνουμε τις συσχετίσεις διαδοχικών φωνημάτων με την χρήση N-grams.
Παρατηρούμε ότι το μοντέλο μας σταματάει στην δημιουργία tri-grams.Είναι λοιπόν πιθανόν η χρήση four-grams να βελτιώσει ακόμη περισσότερο τα αποτελέσματα.
3)Επιπλέον,συχνά στον τομέα αναγνώρισης φωνής χρησιμοποιούνται και άλλες μέθοδοι για την εξαγωγή χαρακτηριστικών,όπως η οπτική αναγνώριση της θέσης και του σχηματισμού των χειλιών και της γλώσσας του ομιλητή (με τοποθέτηση οπτικού αισθητήρα).Συχνά αυτό οδηγεί σε καλύτερη αναγνώριση των φωνημάτων.