Επεξεργασία Φωνής και Φυσικής Γλώσσας
Αναφορά Προπαρασκευής 1ου Εργαστηρίου

Συνεργάτες:

- **Τζίνης Ευθύμιος Α.Μ:03112007**
- **Ευάγγελος Χατζηπανταζής Α.Μ:03112050**

Παρακάτω επεξηγούμε σε όλα τα βήματα που μας ζητήθηκε να υλοποιήσουμε, την διαδικασία που ακολουθήσαμε. Έγινε προσπάθεια προκειμένου να γίνει πιο ευανάγνωστη η διαδικασία και να αυτοματοποιηθεί να γίνει με την χρήση ενός **Makefile*** που περιέχει όλες τις εντολές που χρειάζεται να δώσουμε στο shell χρησιμοποιώντας το openfst. Καθώς επίσης και τις εντολές της python για το compile&run των scripts που φτιάξαμε:

Βήμα1 και 2:

Παραθέτουμε το αρχείο vima1.py στο οποίο περιλαμβάνεται ο κώδικας για τα βήματα 1,2 και η σύνδεση με το βήμα3.Τα αρχεία εισόδου μπορούν να δωθούν είτε ως arguments απο το command line είτε θα ανοίξουν αυτόματα τα αρχεία **train_gr.txt** kai **train_greng.txt** που περιέχονται στον φάκελο. Το πρόγραμμα είναι γραμμένο σε python3.

Στο κομμάτι αυτό που αφορά την εκπαίδευση του μοντέλου συγκρίναμε 2 κείμενα και εξαγάγαμε ορισμένους κανόνες αντιστοίχισης,μεταξύ ελληνικών φθόγκων και της greeklish απόδοση τους.Η ιδέα πίσω από την εκμάθυνση του μοντέλου είναι η εξής:

Σε τρια dictionaries μοιράσαμε τους 1-1,1-2 και 2-1 κανόνες οι οποίοι αντιστοιχίζονταν ανάλογα με τα μήκη των λέξεων προς σύγκριση.Κρατώντας,έτσι προηγούμενους κανόνες και συγκρίνοντας τους καινούργιους με τους ήδη υπάρχοντες,προσθέταμε τους νέους και αυξάναμε τον αριθμό εμφανίσεων των παλιών.Έτσι τα τρια λεξικά,στο τέλος περιείχαν μια αρκετά μεγάλη γκάμα από κανόνες,εκ των οποίων κάποιοι ήταν αναπόφευκτα λάθος.Οστόσο παρατηρήσαμε οτι η συχνότητα των λανθασμένων κανόνων ήταν πολύ μικρότερη σε σχέση με αυτή των σωστών στις περισότερες περιπτώσεις.Για τον λόγο αυτόν,φιλτράραμε τους κανόνες με ένα κατώφλι πιθανότητας εμφάνισης εξαφανίζοντας κάποιους από αυτούς.Το επιλεγμένο κατώφλι ήταν αρκετά μικρό ώστε να μην κόψει αρκετούς παραπανίσιους σωστούς κανόνες.Τέλος,οι λάθος κανόνες που ίσως υπάρχουν στο μοντέλο μπορούν να φιλτραριστούν στην συνέχεια με την χρήση ορθογράφου,γιαυτό ανεχτήκαμε ενα μικρό ποσοστό λάθους.

Βήμα3:

Σε αυτό το βήμα με βάση τους κανόνες που βρήκαμε από τα βήματα 1-2 και τα αντίστοιχα κόστη μετάβασης υλοποιήσαμε έναν μετατροπέα ο οποίος έχει ώς είσοδο τα greeklish και ώς έξοδο τους Ελληνικούς χαρακτήρες που αυτά αντιστοιχίζονται. Όλες οι μεταβάσεις γίνονται πάνω στην ίδια κατάσταση (έστω 0) ή οποία είναι και τελική. Το script που υλοποιήσαμε είναι το **mG.py** και μπορούμε πολύ εύκολα να το εκτελέσουμε και να σχεδιάσουμε το αυτόματο απλά με την εντολή **make G** στο shell. Επειδή το fstdraw δεν τυπώνει στα αρχεία εικόνων ps και jpeg τα ελληνικά μπορούμε απλά να ανοίξουμε το αρχείο dot με το πρόγραμμα **Xdot** και να δούμε τα αποτελέσματα.

Βήμα4:

Σε αυτό το βήμα φτιάχνουμε έναν μετατροπέα που αντιστοιχεί κάθε λατινικό χαρακτήρα στον εαυτό του με πολύ μεγάλο σταθερό κόστος. Μέσα στο script **mI.py** του φακέλου εργασίας μπορούμε να δούμε την δημιουργία ενός **alphabet.syms** που έχει όλη την αγγλική αλφάβητο (δημιουργείται μέσα στο script) και χρησιμεύει στην δημιουργία του αρχείου **I.txt** που θα κάνουμε fstcompile.

Αυτός ο μετατροπέας χρησιμεύει στην διαδικασία μετατροπής greeklish->Ελληνικά ώς εξής: Μέσα στο κείμενο που θα μας δωθεί να μεταφράσουμε μπορεί να υπάρχουν κάποιες λέξεις που είναι αγγλικές και ορθογράφος δεν θα χρειαστεί να αντιστοιχήσει σε ελληνικές. Άρα μπορούμε να περάσουμε το αρχικό μας κείμενο πρώτα από έναν Αγγλικό ορθογράφο ούτως ώστε μία λέξη που υπάρχει στο αγγλικό λεξικό να μην χρειάζεται μετάφραση στα ελληνικά. Το μεγάλο σταθερό κόστος χρησιμεύει ουσιαστικά στο να κάνει όσο μεγαλύτερη την πιθανότητα μετάβασης στην προσπάθεια αντιστοίχησης με το αγγλικό λεξικό.

Βήμα5:

Στο βήμα αυτό υλοποιούμε έναν αποδέκτη που δέχεται όλες τις ελληνικές λέξεις που υπάρχουν στο λεξικό που μας δόθηκε (χρησιμεύει στα τελικά στάδια του αρθογράφου). Παίρνουμε το κείμενο **el_caps_noaccent.dict** μέσω του script **A1.py** και φτιάχνουμε τα αρχεία:

- **A1.txt:** Περιέχει όλες τις δυνατές μεταβάσεις από χαρακτήρα σε χαρακτήρα μέχρι την αποδοχή της τελικής λέξης. Άρα έχουμε για κάθε λέξη θα έχουμε αρχικά ένα κλαδί που ξεκινάει από την αρχική κατάσταση και καθώς παίρνουμε έναν έναν τους χαρακτήρες σε ελληνικά, αποδεχόμαστε ή όχι την λέξη.
- **dict.txt:** Περιέχει μια αντιστοίχηση των ελληνικών γραμμάτων με integers προκειμένου να μπορέσουμε να χρησιμοποιήσουμε το fst.

Όπως καταλαβαίνουμε μέχρι εδώ έχουμε υλοποιήσει ένα NFA αφού από την αρχική κατάσταση με το ίδιο γράμμα ως είσοδο μπορώ να μεταβώ σε πάνω από μία καταστάσεις. Μπορούμε να εκτελέσουμε στο shell **make A1py A11** για την μέχρι εδώ διαδικασία.

Βήμα6:

Σε αυτό το βήμα παίρνουμε έτοιμο το προηγούμενο NFA και το κάνουμε ντετερμινιστικό και το μειώνουμε χρησιμοποιώντας τις εντολές του openfst: fstdeterminize, fstminimize. Μπορούμε να εκτελέσουμε την όλη διαδικασία απλά
με: **make Adet Amin**. Γνωρίζουμε πόσο μεγάλο είναι το fsm που υλοποιήσαμε και γι αυτό δεν το σχεδιάζουμε καθόλου. Παρόλα αυτά για λόγους πληρότητας έχουμε ενισχύσει το Makefile με την δυνατότητα στόχου **make Amindraw** που το σχεδιάζει.

Προκειμένου να μην εκτελούμε τους επιμέρους στόχους της διαδικασίας του A1 (βήματα 5 και 6) μπορούμε απλά να εκτελέσουμε **make A1**.

Πόσο μεγάλο είναι το **A1.fst**;
-->Χρησιμοποιούμε την εντολή fstinfo

```
thymios@thymios:~/Desktop/lab1.SpeechProc$ fstinfo A1.fst
fst type                                    vector
arc type                                    standard
input symbol table                          none
output symbol table                         none
# of states                                 4212180
# of arcs                                   4212179
initial state                               0
# of final states                           393948
# of input/output epsilons                  345672
# of input epsilons                         345672
# of output epsilons                        345672
input label multiplicity                    3272.43
output label multiplicity                   3272.43
# of accessible states                      4212180
# of coaccessible states                    4212180
# of connected states                       4212180
# of connected components                   1
# of strongly conn components               4212180
```

Βήμα7:

Με ακριβώς την ίδια λογική με τα προηγούμενα βήματα (5-6) κάνουμε για το αγγλικό λεξικό. Επομένως απλά παίρνουμε το αρχείο **en_caps_noaccent.dict** μέσω του script **A2.py** και δημιουργούμε τα αντίστοιχα αρχεία (βλέπε βήμα 5):

- **A2.txt**
- **dicten.txt**

Κάνουμε την ίδια διαδικασία με το βήμα 6 για το **A2.fst** που δημιουργήσαμε.

Μπορούμε να δούμε την όλη διαδικασία να τρέχει χωρίς να εκτελούμε κάθε επιμέρους στόχο διαφορετικά απλά με την εκτέλεση στο shell της εντολής: **make A2**.

Για τους ίδιους λόγους με πριν δεν ζωγραφίζουμε την εικόνα αλλά παρόλα αυτά υπάρχει ώς επιλογή στόχου στο **Makefile**.

Μπορούμε να εκτελέσουμε όλα τα βήματα απλά πατώντας **make** ή **make all**.

Κώδικας του Makefile:

```
all: G I A1 A2
G: Gpy G1 G2
Gpy:
	python3 vima1.py
G1:
	fstcompile --isymbols=inputG --osymbols=outputG G.txt G.fst
G2:
	fstdraw --isymbols=inputG --osymbols=outputG G.fst G.dot|dot -Tps G.dot >G.ps
	fstprint --isymbols=inputG --osymbols=outputG G.fst
I: Ipy I1 I2
Ipy:
	python3 mI.py
I1:
	fstcompile --isymbols=alphabet.syms --osymbols=alphabet.syms I.txt I.fst
I2:
	fstdraw --isymbols=alphabet.syms --osymbols=alphabet.syms I.fst I.dot|dot -Tps I.dot >I.ps
I: A1py A11 Adet Amin
A1py:
	python3 A1.py
A11:
	fstcompile --isymbols=dict.txt --osymbols=dict.txt A1.txt A1.fst
Adet:
	fstdeterminize A1.fst >A1-det.fst
Amin:
	fstminimize A1-det.fst min-A1.fst
Amindraw:
	fstdraw --isymbols=dict.txt --osymbols=dict.txt min-A1.fst A1.dot|dot -Tps A1.dot >A1.ps
A2: A2py A21 A2det A2min
A2py:
	python3 A2.py
A21:
	fstcompile --isymbols=dicten.txt --osymbols=dicten.txt A2.txt A2.fst
A2det:
	fstdeterminize A2.fst >A2-det.fst
A2min:
	fstminimize A2-det.fst min-A2.fst
A2mindraw:
	fstdraw --isymbols=dicten.txt --osymbols=dicten.txt min-A2.fst A2.dot|dot -Tps A2.dot >A2.ps
```