



ΣΤΡΑΤΙΩΤΙΚΗ ΣΧΟΛΗ
ΕΥΕΛΠΙΔΩΝ

Τμήμα Στρατιωτικών
Επιστημών

ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ

ΔΙΔΡΥΜΑΤΙΚΟ ΔΙΑΤΜΗΜΑΤΙΚΟ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΑΚΑΔΗΜΑΪΚΟΥ ΕΤΟΥΣ 2016-17

ΣΧΕΔΙΑΣΗ & ΕΠΕΞΕΡΓΑΣΙΑ

ΣΥΣΤΗΜΑΤΩΝ (SYSTEMS ENGINEERING)

(ΠΔ 96 /2015/ΦΕΚ 163Α'/20.08.2014)



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

Σχολή Μηχανικών Παραγωγής & Διοίκησης

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΑΤΡΙΒΗ

Υλοποίηση συστήματος αναγνώρισης φωνητικών εντολών στην Ελληνική γλώσσα

Διατριβή που υπεβλήθη για την μερική ικανοποίηση των απαιτήσεων για
την απόκτηση Μεταπτυχιακού Διπλώματος Ειδίκευσης

Υπό:

ΦΩΤΙΟΣ ΚΛΕΑΡΧΟΥ ΠΑΝΤΑΖΟΓΛΟΥ

A.M.: 2015018010

ΟΚΤΩΒΡΙΟΣ 2017

Η Μεταπτυχιακή Διατριβή του κυρίου Πανταζόγλου Φωτίου εγκρίνεται:

ΤΡΙΜΕΛΗΣ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

Επίκουρος Καθηγητής Δρ. Παπαδάκης Νικόλαος(Επιβλέπων),.....

Καθηγητής Δρ. Δάρας Νικόλαος ,.....

Επίκουρος Καθηγητής Δρ. Κλάδης Γεώργιος ,.....

ΣΕΛΙΔΑ ΣΚΟΠΙΜΑ ΚΕΝΗ

© Copyright υπό
Φώτιος Πανταζόγλου
Έτος 2017

Στην Μαρία την Ελευθερία και τον Νίκο

ΣΕΛΙΔΑ ΣΚΟΠΙΜΑ ΚΕΝΗ

ΕΥΧΑΡΙΣΤΙΕΣ

Η Μεταπτυχιακή διατριβή που ακολουθεί αποτελεί για μένα το επιστέγασμα μιας συνεπούς και διαρκούς προσπάθειας που ξεκίνησε περίπου πριν δύο χρόνια. Μια προσπάθεια, κατά την διάρκεια της οποίας προσπάθησα να φανώ αντάξιος των προσδοκιών της οικογένειας μου αλλά και των καθηγητών μου. Πρέπει να ομολογήσω ότι ξεκίνησα την φοίτηση μου με πίστη στον εαυτό μου, αλλά και προβληματισμό. Προβληματισμό για τις προκλήσεις που θα αντιμετώπιζα κατά την διάρκεια της φοίτησης μου. Έχοντας να ισορροπήσω ανάμεσα στις οικογενειακές, τις εργασιακές, αλλά και τις φοιτητικές μου υποχρεώσεις αφέθηκα να συμπαρασυρθώ σε αυτό το καινούργιο, που ανοιγόταν μπροστά μου. Νέοι άνθρωποι, νέοι εξαιρετικοί δάσκαλοι, νέες προκλήσεις, νέα γνωστικά αντικείμενα συνέθεταν πλέον την νέα καθημερινότητά μου. Ομολογώ ότι για μένα τουλάχιστον η όλη διαδικασία ήταν καθόλα παιδευτική και εκπαιδευτική. Έμαθα πολλά, αγάπησα ξανά την έρευνα έβαλα ξανά νέους στόχους για τις αναζητήσεις μου, στο κόσμο της επιστήμης και της γνώσης.

Φυσικά όλα αυτά δεν θα ήταν πραγματικότητα χωρίς την βοήθεια των εξαιρετικών δασκάλων που γνώρισα κατά την διάρκεια της φοίτησης μου. Θα ήθελα να τους ευχαριστήσω όλους, για τον ζήλο και την υπομονή απέναντί μου, που επέδειξαν όλο αυτό τον καιρό.

Δεν θα μπορούσα να ξεκινήσω την προσπάθειά μου, αν δεν με είχαν βοηθήσει δυο εξαιρετικοί δάσκαλοι, καθηγητές, και άνθρωποι. Οι κύριοι Αναστάσιος Ελευθερίου και Αναστάσιος Τσελεπίδης στήριξαν την προσπάθειά μου από το ξεκίνημα της. Τους ευχαριστώ από καρδιάς.

Ένα μεγάλο ευχαριστώ δικαιωματικά θα ήθελα να απευθύνω και προς τους συμφοιτητές μου (με αλφαβητική σειρά) Εύα Αλεξοπούλου, Γιώργο Γαϊτανάκη και Γιάννη Γιαβάσογλου. Αποτέλεσαν αστείρευτη πηγή δύναμης, χιούμορ και έμπνευσης για μένα αυτά τα χρόνια και σίγουρα με ωθούσαν κάθε φορά να υπερβώ τον εαυτό μου. Τους οφείλω πολλά.

Η Μεταπτυχιακή αυτή διατριβή οφείλει πολλά στον κύριο Nickolay Shmyrev, υπεύθυνο ανάπτυξης και υποστήριξης της πλατφόρμας CMU Sphinx. Ο Nickolay συνέβαλε αποφασιστικά με την εμπειρία του και τον ενθουσιασμό του κατά την φάση της εκπαίδευσης και παραμετροποίησης του Ελληνικού μοντέλου.

Πολλά ευχαριστώ για την φίλη ηθοποιό κυρία Λίλα Μουτσοπούλου που είχε την καλοσύνη να μου παραχωρήσει προσωπικές της ηχογραφήσεις, για τις ανάγκες εκπαίδευσης του Ελληνικού μοντέλου, αλλά και να φροντίσει ώστε να έχω στην διάθεσή μου υλικό υψηλής ηχητικής ποιότητας.

Κατά την επιμέλεια της συγγραφής της Μεταπτυχιακής διατριβής, πολύτιμη υπήρξε η συμβολή της συναδέλφου μου στο ΕΛ.ΚΕ.Θ.Ε. κυρίας Κυριακής Μυλωνά, την οποία και ευχαριστώ για τις πολύ χρήσιμες συμβουλές και παρατηρήσεις της.

Θα μπορούσα να έγραφα πάρα πολλά για τους εξαιρετικούς επιβλέποντες καθηγητές μου, κυρίους Παπαδάκη, Δάρα και Κλάδη. Θέλω να αρκесτώ απλά σε μεγάλο ευχαριστώ για την δυνατότητα που μου έδωσαν να μπορέσω να δουλέψω πάνω σε κάτι πραγματικά πρωτοποριακό και να επεκτείνω τους γνωσιαιούς μου ορίζοντες. Πιστεύω ότι το επιστημονικό αποτέλεσμα της Μεταπτυχιακής διατριβής δικαίωσε την επιλογή τους, και την στήριξη τους προς το πρόσωπο μου. Είναι δεδομένο ότι οι επιστημονικές μου αναζητήσεις πλέον, θα μου επιτρέψουν να έχω και στην συνέχεια συνεργασίες μαζί τους.

Όλη αυτή η προσπάθεια δεν θα γινόταν, μάλλον δεν θα ξεκινούσε καν αν δεν υπήρχε η στήριξη της οικογένειάς μου. Σε πρώτη φάση με ώθησαν να αιτηθώ της εισαγωγής μου στο Μεταπτυχιακό Πρόγραμμα Σπουδών. Στην συνέχεια και κατά την διάρκεια των σπουδών μου, και κατανόησαν τις απαιτήσεις που αυτές είχαν σε προσωπικό χρόνο και με στήριζαν καθημερινά με τον δικό τους μοναδικό και ξεχωριστό τρόπο. Η συμβολή και στήριξη της συζύγου μου Μαρίας και των παιδιών μου Ελευθερίας και Νίκου ήταν καθοριστική για την επιτυχή ολοκλήρωση των Μεταπτυχιακών μου σπουδών. Χωρίς αυτούς απλά δεν θα είχα το καθημερινό κίνητρο για παραπάνω δουλειά και προσπάθεια. Ξέρουν πόσο τους αγαπάω και ότι ένα απλό ευχαριστώ δεν μπορεί να αποδώσει το τι αισθάνομαι για αυτούς.

Όλη μου η προσπάθεια αλλά και η Μεταπτυχιακή μου διατριβή είναι αφιερωμένη σε αυτούς.

Φώτιος Κλ. Πανταζόγλου
Ηράκλειο-Οκτώβριος 2017

ΣΕΛΙΔΑ ΣΚΟΠΙΜΑ ΚΕΝΗ

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΕΡΙΛΗΨΗ	1
ΕΙΣΑΓΩΓΗ	3
§1 Γενικά	3
§2 Ορισμοί Βασικές έννοιες	7
ΚΕΦΑΛΑΙΟ 1	10
Επικοινωνία Λόγος Αυτόματη Φωνητική Αναγνώριση	10
§1 Επικοινωνία. Από την αρχή της ιστορίας μέχρι σήμερα	10
§2 Προφορικός και Γραπτός λόγος	19
§3 Αυτόματη Φωνητική Αναγνώριση	24
3.1 Γενικά	24
3.2 Εφαρμογές	25
3.3 Αρχιτεκτονική και φιλοσοφία σχεδίασης	28
3.4 Κατηγορίες συστημάτων φωνητικής αναγνώρισης	30
3.5 Τεχνικές μοντελοποίησης	31
3.5.1 Η ακουστικό-φωνητική προσέγγιση	31
3.5.2 Προσέγγιση με βάση την αναγνώριση μοτίβου	31
3.5.3 Προσέγγιση με βάση την αναγνώριση προτύπων	32
3.5.4 Δυναμική χρονική στρέβλωση (Dynamic time warping-DTW)	32
3.5.5 Προσέγγιση με βάση την υπαρκτή γνώση	32
3.5.6 Προσέγγιση με βάση στατιστικά μοντέλα	32
3.5.7 Εφαρμογές με βάση την εκπαίδευση	33
3.5.8 Προσέγγιση με βάση την Τεχνητή Νοημοσύνη	33
3.5.9 Στοχαστική προσέγγιση	33
3.5.10 Προσέγγιση με βάση τα Βαθιά Νευρωνικά Δίκτυα (DNN Deep Neural Networks)	33
ΚΕΦΑΛΑΙΟ 2	35
Οι πλατφόρμες αυτόματης φωνητικής αναγνώρισης CMU Sphinx και pocketshinx	35
§1 Γενικά	35
§2 Δομή και βασικά χαρακτηριστικά του προφορικού λόγου	37
2.1 Γενικά	37
2.2 Η δομή και τα χαρακτηριστικά του προφορικού λόγου	37
2.3 Η μεθοδολογία αναγνώρισης του CMU Sphinx	39

§3	Αναλυτική περιγραφή αρχών λειτουργίας και αρχιτεκτονικής CMU Sphinx και Pocketsphinx	41
3.1	Αρχιτεκτονική λειτουργίας του CMU Sphinx	41
3.1.1	Το Front-end μπλοκ	42
3.1.2	Ο αποκωδικοποιητής (decoder)	43
3.1.2.1	Διαχειριστής αναζήτησης	43
3.1.2.2	Γλωσσολόγος	43
3.1.2.3	Ακουστικός βαθμολογητής	43
3.2	Αρχιτεκτονική λειτουργίας του Pocketsphinx	44
3.2.1	Ακουστική μοντελοποίηση	45
3.2.2	Αποκωδικοποιητής	46
3.2.2.1	Αναζήτηση κατά προσέγγιση με βάση δέντρο λέξεων (fwdtree)	46
3.2.2.2	Αναζήτηση κατά προσέγγιση με βάση ένα επίπεδο λεξιλόγιο (fwdflat)	46
3.2.2.3	Αναζήτηση με την δημιουργία πλέγματος (lattice generation)	46
ΚΕΦΑΛΑΙΟ 3		48
Δημιουργία Ελληνικού Μοντέλου για το CMU Sphinx		48
§1	Γενική περιγραφή του Ελληνικού Μοντέλου για το CMU Sphinx	48
§2	Διαδικασίες δημιουργίας Ελληνικού Μοντέλου	50
2.1	Δημιουργία Ελληνικού Γλωσσικού μοντέλου (Language Model)	50
2.2	Δημιουργία Ελληνικού φωνητικού λεξικού (Phonetic dictionary)	55
2.2.1	Δημιουργία λεξιλογίου	56
2.2.2	Δημιουργία μηχανισμού μετατροπής λέξεων σε σύνολο φωνημάτων	56
2.3	Δημιουργία Ελληνικού Ακουστικού μοντέλου (Acoustic Model)	58
2.3.1	Εγκατάσταση απαραίτητου λογισμικού (sphinxtrain, training base)	59
2.3.2	Προετοιμασία ηχητικού υλικού εκπαίδευσης	60
2.3.3	Παραμετροποίηση μηχανισμού εκπαίδευσης	61
2.3.4	Τροφοδοσία sphinxtrain με υλικό εκπαίδευσης	62
ΚΕΦΑΛΑΙΟ 4		65
Σενάρια χρήσης		65
§1	Χειρισμός βραχιόνων σε αυτόνομα υποβρύχια οχήματα (ROV)	65
§2	Εφαρμογές σε περιβάλλοντα στρατιωτικού χαρακτήρα	72
ΚΕΦΑΛΑΙΟ 5		76
Συμπεράσματα		76

ΒΙΒΛΙΟΓΡΑΦΙΑ	<u>79</u>
ΙΣΤΟΤΟΠΟΙ	<u>86</u>
ΠΑΡΑΡΤΗΜΑ	<u>88</u>
1 Δημιουργία γλωσσικού μοντέλου	<u>88</u>
1.1 Κώδικας μακροεντολής breakLines	<u>88</u>
1.2 Κώδικας μακροεντολής CleanEmAll	<u>88</u>
1.3 Ενδεικτικό γλωσσικό μοντέλο test.lm	<u>91</u>
2 Δημιουργία φωνητικού λεξικού	<u>99</u>
2.1 Κώδικας μακροεντολής create_phoneme	<u>99</u>
3 Δημιουργία ακουστικού μοντέλου	<u>112</u>
3.1 Κείμενο προς ηχογράφηση	<u>112</u>
3.2 sphinx_train.cfg	<u>113</u>
4 Δημιουργία Ελληνικού μοντέλου για χρήση σε ROV	<u>119</u>
4.1 Φράσεις που χρησιμοποιήθηκαν για την εκπαίδευση του μοντέλου	<u>119</u>
4.2 Αποτελέσματα εκπαίδευσης Ελληνικού μοντέλου προσαρμοσμένο για χρήση σε εργασίες ROV	<u>120</u>
ΛΙΣΤΑ ΕΙΚΟΝΩΝ-ΣΧΕΔΙΑΓΡΑΜΜΑΤΩΝ	<u>128</u>

ΠΕΡΙΛΗΨΗ

Με την υπερβολική επίδοση του ανθρώπου, κατά τους νεότερους χρόνους, στις λεγόμενες θετικές επιστήμες παρατηρήθηκε και ανάλογη δεσπόζουσα ανάπτυξη της τεχνολογίας. Η τεχνολογία είναι γέννημα της ανάγκης να αποδεσμευτεί ο άνθρωπος από την κυριαρχία της φύσης αλλά και να καλυτερέψει τις συνθήκες ζωής του. Η τεχνολογία, σαν αδιάκοπη συνέχεια και πορεία από σταθμό σε σταθμό και από κατάρκτηση σε κατάρκτηση, όπου η νεότερη προϋποθέτει την προγενέστερη, κατάληξε να πάρει τις διατάσεις που έχει σήμερα. Είναι η βάση της οικονομίας τόσο, που η οικονομική ανάπτυξη των υπανάπτυκτων χωρών να ταυτίζεται με την τεχνολογική τους πρόοδο. Έγινε κυρίαρχο στοιχείο στην ζωή του ανθρώπου, γιατί με αυτήν είδε να πολλαπλασιάζονται οι δυνατότητες του και πίστεψε ότι με την βοήθεια της θα ανοιγόταν μπροστά του ένας καινούργιος δρόμος σαφώς πιο βελτιωμένος και ποιοτικός από τον δύσκολο δρόμο που είχε ακολουθήσει μέχρι εκείνη την στιγμή. Οι αλληπάλληλες τεχνολογικές επαναστάσεις άλλαξαν ριζικά την μορφή του κόσμου. Δημιούργησαν νέα δεδομένα, οικονομικά, κοινωνικά, πολιτικά και δημογραφικά.

Η Μεταπτυχιακή διατριβή που παρουσιάζεται σε αυτή την συγγραφή, εμπνεύστηκε από την ανάγκη του συγγραφέα να βρει τεχνική λύση στο πρόβλημα ελέγχου μιας μηχανικής υποδομής μέσω απλών εντολών. Είτε ο έλεγχος αφορά μηχανικά μέρη ενός αυτόνομου υποβρυχίου οχήματος, είτε αφορά μηχανικά μέρη ενός έξω-σκελετού υποστήριξης είτε αφορά μια οποιαδήποτε άλλη μηχανική εφαρμογή η ανάγκη για απλό άμεσο και ακριβή έλεγχο είναι υπαρκτή και απαιτητή σε μια πληθώρα εφαρμογών.

Στην καθημερινότητα μας ο πιο απλός και άμεσος τρόπος για να ασκούμε έλεγχο και να δίνουμε εντολές είναι ο προφορικός λόγος. Η εκφορά του προφορικού λόγου χαρακτηρίζει όλα τα νοήμονά όντα και συνεπώς η ιδέα να χρησιμοποιηθεί προφορικός λόγος ώστε να ελέγξει κάποιος μια μηχανική κατασκευή ή έναν μικροϋπολογιστή δεν είναι καινούργια. Αλλά για να φτάσει κάποιος στην διαδικασία να μπορεί να ασκήσει έλεγχο θα πρέπει να έχει προηγηθεί το πρώτο στάδιο της κατανόησης και αναγνώρισης του εκφερόμενου προφορικού λόγου. Θα πρέπει να έχει προηγηθεί η αυτόματη αναγνώριση φωνής. (ASR - Automatic Speech Recognition)

Μια από τις πιο εξέχουσες προσπάθειες ανοιχτού κώδικα στα συστήματα ASR είναι και το CMU Sphinx. Η πλατφόρμα αυτόματης φωνητικής αναγνώρισης CMU Sphinx έχει αναπτυχθεί από κοινού από το Πανεπιστήμιο Carnegie Mellon τα εργαστήρια της Sun Microsystems και τα ερευνητικά εργαστήρια της Mitsubishi (Lamere et al., 2003). Έχει «χτιστεί» στην κυριολεξία με την γλώσσα προγραμματισμού Java και αποτελεί μια πολύ

ευέλικτη πλατφόρμα που υποστηρίζει όλους τους τύπους ακουστικών μοντέλων που είναι βασισμένοι σε Κρυφά Μαρκοβιανά Μοντέλα (HMM). Παρότι υποστηρίζει τις πιο γνωστές παγκοσμίως γλώσσες, δεν είχε μέχρι πρότινος υποστήριξη για την Ελληνική γλώσσα, καθώς απουσίαζε το Ελληνικό μοντέλο της πλατφόρμας.

Έχοντας γνώση της παραπάνω πραγματικότητας, ο συγγραφέας εργάστηκε στα πλαίσια της Μεταπτυχιακής του διατριβής με στόχο την δημιουργία του Ελληνικού μοντέλου. Η Μεταπτυχιακή διατριβή έρχεται να συνδράμει στην παγκόσμια προσπάθεια δημιουργίας εργαλείων αναγνώρισης φωνής, παρουσιάζοντας και παραδίδοντας προς την κοινότητα το Ελληνικό Μοντέλο, το οποίο αποτελεί πλέον μέρος της υπολογιστικής πλατφόρμας CMU Sphinx.

Τα τελικά επιστημονικά αποτελέσματα που προέκυψαν από τις εργασίες κατά την διάρκεια της Μεταπτυχιακής διατριβής, είναι ήδη ελεύθερα διαθέσιμα προς τους ενδιαφερόμενους χρήστες από τις παρακάτω διευθύνσεις του διαδικτύου (αποθετήρια Στρατιωτικής Σχολής Ευελπίδων και CMU Sphinx):

<https://gitlab.sse.gr/fpantazoglou/omilia>

<https://goo.gl/9v3QqG>

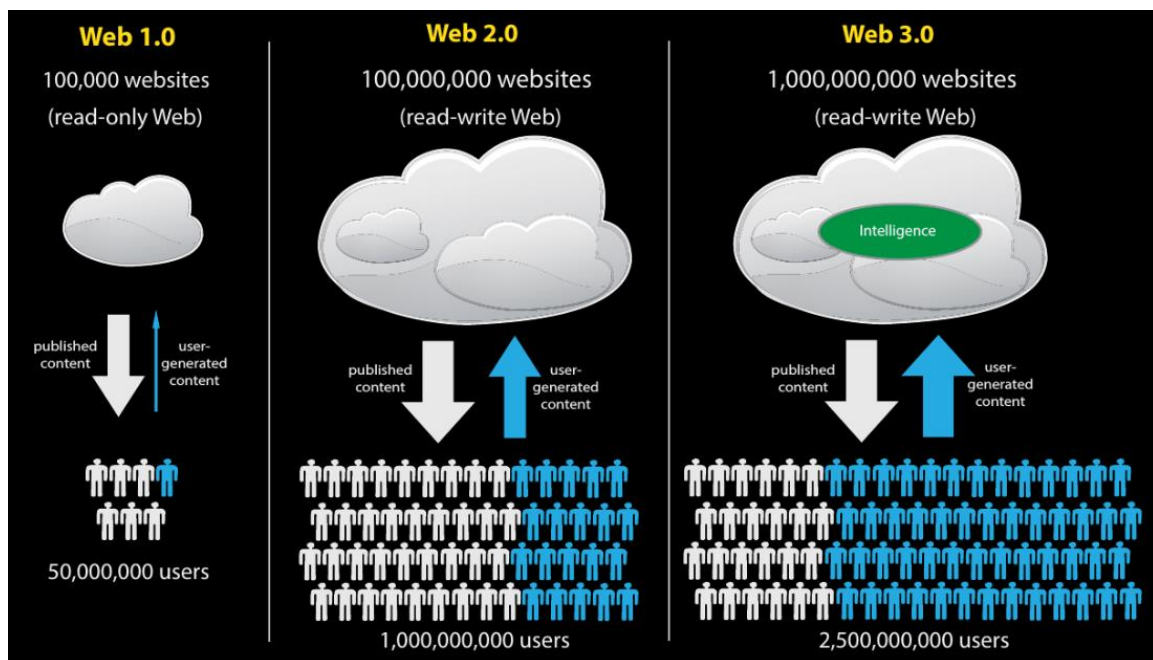
Είναι βαθιά πεποίθηση του συγγραφέα αλλά και των επιβλεπόντων καθηγητών του, ότι η ύπαρξη του Ελληνικού μοντέλου για την πλατφόρμα CMU Sphinx, θα φανεί ιδιαίτερα χρήσιμη στους Έλληνες χρήστες καθώς θα τους επιτρέψει να ενσωματώσουν την Ελληνική γλώσσα σε εφαρμογές που εμπεριέχουν την τεχνολογία της αυτόματης φωνητικής αναγνώρισης.

ΕΙΣΑΓΩΓΗ

§1. Γενικά

Η εποχή μας χαρακτηρίζεται από την ραγδαία εξάπλωση νέων τεχνολογιών και πρακτικών. Με κύριο στόχο την βελτίωση της καθημερινής ζωής του ανθρώπου αλλά και την επέκταση των ορίων της υφιστάμενης γνώσης στα πεδία της επιστήμης, νέες πρωτοποριακές τεχνολογίες έρχονται να συνδράμουν στην αέναη αυτή προσπάθεια του ανθρώπου. Σημαντικό εργαλείο στις προσπάθειες μας αποτελούν τόσο οι εξελίξεις στον τομέα των μικρών υπολογιστών αλλά και η πρόοδος που σημειώνεται στις γλώσσες προγραμματισμού. Έτσι συσκευές χαμηλού κόστους, όπως το [Arduino](#) ή το [Raspberry Pi 3](#), υποστηριζόμενες από γλώσσες προγραμματισμού ανοιχτού κώδικα έχουν προσδώσει τεράστια ώθηση στην δημιουργικότητα μας και στην ανάπτυξη νέων εφαρμογών που δίνουν λύσεις σε μια σειρά απλών η και σύνθετων προβλημάτων.

Ταυτόχρονα ένα διαρκώς βελτιούμενο ,αλλά και ραγδαία αναπτυσσόμενο τοπίο στον χώρο του διαδικτύου (Εικόνα 1), μας επιτρέπει να ανταλλάσσουμε πληροφορίες, να



Εικόνα 1. .Η ανάπτυξη του διαδικτύου μέσα από αριθμούς (<https://goo.gl/pUX77U>)

ενημερωθούμε για νέες εξελίξεις, να συνεργαστούμε πάνω σε κοινά προβλήματα με άτομα από όλο τον κόσμο.

Η ανοιχτή πρόσβαση σε πληροφορίες και λύσεις έχει βοηθήσει σημαντικά στην πιο γρήγορη αντιμετώπιση προβλημάτων και εξεύρεσής λύσεων. Διαδικτυακοί τόποι ελεύθερης πρόσβασης όπως το [Sourceforge](#) το [Stack Overflow](#) το [GitHub](#) ή το [w3schools](#) αποτελούν χαρακτηριστικά παραδείγματα ιστότοπων όπου χιλιάδες χρήστες καθημερινά ανταλλάσσουν απόψεις, λύνουν προγραμματιστικά προβλήματα, συνεισφέρουν σε κώδικα ανοιχτής πρόσβασης, συμβάλλοντας ο καθένας με τον τρόπο του στην ευρύτερη διάδοση της γνώσης.

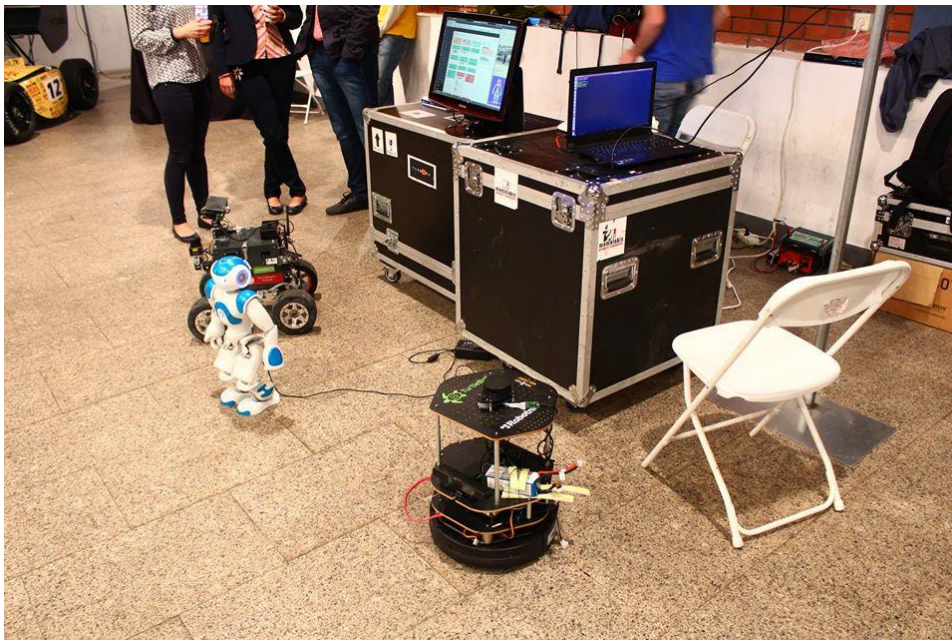
Ένα από τα πιο παλιά ζητήματα με τα οποία έχει ασχοληθεί η επιστημονική κοινότητα είναι η αυτόματη αναγνώριση της ανθρώπινης (ASR) φωνής από τεχνικά συστήματα. Όπως αναφέρουν οι Martin, Zadell, Grunza, Hmcher, & Reddy, (1974), ήδη από την δεκαετία του 1950 είχαν κατασκευαστεί συστήματα, που σκοπό είχαν την αναγνώριση της ανθρώπινης φωνής. Προφανώς μιλάμε για τα προγενέστερα στάδια της όλης διαδικασίας. Μια διαδικασία που άρρηκτα συνδεδεμένη με την εξέλιξη των υπολογιστικών προγραμμάτων των μικρό επεξεργαστών αλλά και του διαδικτύου πλέον. Η επιτυχημένη αναγνώριση της ανθρώπινης φωνής οδηγεί με την σειρά της, σε έναν αριθμό χρήσιμων εφαρμογών που μπορεί να είναι από πολύ απλές (φωνητική αναγνώριση ατόμου) μέχρι και πιο πολύπλοκές (χειρισμός μηχανικών συστημάτων).

Μια από τις πολλές προσπάθειες που λαμβάνει χώρα τα τελευταία χρόνια εκπορεύεται από την συνεργασία του [Carnegie Mellon University](#) των [Sun Microsystems Laboratories](#) και των [Mitsubishi Electric Research Laboratories](#) έχοντας αποδώσει σαν καρπούς μια πλατφόρμα εργαλείων υπό το όνομα CMU Sphinx (Lamere et al., 2003). Ένα από τα εργαλεία της πλατφόρμας CMU Sphinx είναι και το Pocketsphinx (Huggins-Daines et al., 2006) που αποτελεί μια έκδοση που είναι σχεδιασμένη να είναι φιλική για συσκευές χειρός και κινητά τηλέφωνα, χωρίς να υστερεί σε απόδοση όταν «τρέχει» σε κλασικά υπολογιστικά συστήματα. Αυτό το χαρακτηριστικό του Pocketsphinx (χρήση μικρών υπολογιστικών πόρων) το έχει κάνει ιδιαίτερα αγαπητό ανάμεσα στους χρήστες του CMU Sphinx. Για τον λόγο αυτό και προτιμήθηκε από τον συγγραφέα σαν πλατφόρμα ανάπτυξης για τις εργασίες της Μεταπτυχιακής διατριβής του, καθώς και οι υπολογιστικοί πόροι που είχε στην διάθεση του ήταν περιορισμένοι, αλλά και το χρονικό πλαίσιο της όλης διαδικασίας αυστηρά καθορισμένο.

Το Pocketsphinx όπως και το CMU Sphinx συνολικά, υποστηρίζουν διάφορες γλώσσες (Αγγλικά, Γερμανικά, Γαλλικά, Ισπανικά, Ρωσικά κτλ.) που μας επιτρέπουν να έχουμε μια σειρά υπηρεσιών ASR. Για παράδειγμα τον προφορικό λόγο να τον μετατρέψουμε σε γραμμές

γραπτού λόγου, η να «εκπαιδεύσουμε» το ακουστικό μας μοντέλο σύμφωνα με τις ανάγκες μας και να εντάξουμε τις υπηρεσίες αυτές σε ρομποτικά συστήματα.

Δυστυχώς όμως μέχρι σήμερα δεν υπήρχε υποστήριξη για την Ελληνική γλώσσα στο διεθνές αποθετήριο κώδικα του προγράμματος. Υπάρχουν βέβαια σημαντικές προσπάθειες (Εικόνα 2) που παρουσιάζονται αναλυτικά στην εργασία των Tsardoulis, Symeonidis, & Mitkas, (2015), όπου έχει χρησιμοποιηθεί η πλατφόρμα του CMU Sphinx για τις ανάγκες του Ευρωπαϊκού ερευνητικού προγράμματος RAPP (Robotic Applications for Delivering Smart User Empowering Applications <http://rapp-project.eu/>), πλην όμως καλύπτοντας ένα περιορισμένο και αυστηρά καθορισμένο αριθμό λέξεων της Ελληνικής γλώσσας.



Εικόνα 2. Χρήση του CMU Sphinx για τις ανάγκες του Ευρ. Ερευνητικού προγράμματος RAPP (<http://rapp-project.eu/>)

Η Ελληνική γλώσσα έχει τα δικά της γλωσσολογικά χαρακτηριστικά. Για να είναι εφικτή η δημιουργία του Ελληνικού μοντέλου ήταν απαραίτητη η εκ του μηδενός δημιουργία του. Έχοντας σαν οδηγό τις γενικές αρχές δημιουργίας ενός νέου μοντέλου, όπως αυτές παρουσιάζονται στον ιστότοπο υποστήριξης του CMU Sphinx (<https://goo.gl/ocfTLT>), και ακολουθώντας συγκεκριμένη μεθοδολογία, που θα παρουσιαστεί αναλυτικά, έγινε εφικτό να προστεθεί και η Ελληνική γλώσσα στις γλώσσες που πλέον υποστηρίζονται από την

υπολογιστική πλατφόρμα εργαλείων CMU Sphinx. Θα πρέπει εδώ να τονιστεί ότι πρόκειται για μια συνεχή διεργασία. Έχοντας στην διάθεση μας νέο υλικό, που μπορούμε να χρησιμοποιήσουμε στην διαδικασία εκπαίδευσης του μοντέλου, μπορούμε συνεχώς να προσαρμόζουμε το μοντέλο μας ώστε αυτό να ανταποκρίνεται καλύτερα στις απαιτήσεις μας. Ταυτόχρονα το εξελίσσουμε αλλά και βελτιώνουμε την ακρίβεια αλλά και ταχύτητα λειτουργίας του.

Τέλος η δημιουργία του Ελληνικού ακουστικού μοντέλου μας επέτρεψε επίσης να έχουμε στην διάθεση μας ένα εργαλείο μετατροπής των Ελληνικών λέξεων σε μια διεθνή μορφή που γίνεται αντιληπτή από το πρόγραμμα του CMU Sphinx, και ιδιαίτερα το Pocketsphinx που χρησιμοποιήσαμε. Αυτό αποτελεί και μια παρακαταθήκη δουλειάς για μελλοντικές προσπάθειες βελτιστοποίησης αλλά και επέκτασης της ήδη υπάρχουσας προσπάθειας. Ο ενδιαφερόμενος δυνητικός χρήστης, του Ελληνικού μοντέλου για το CMU Sphinx, μπορεί με τα παρεχόμενα αποτελέσματα της Μεταπτυχιακής διατριβής να συμπληρώσει, επεκτείνει, βελτιώσει την ήδη υπάρχουσα προσπάθεια.

Η Μεταπτυχιακή διατριβή έχει χωριστεί σε πέντε κεφάλαια.

- Το πρώτο κεφάλαιο πραγματεύεται τα θέματα της επικοινωνίας του λόγου και της αυτόματης φωνητικής αναγνώρισης, κάνοντας ταυτόχρονα και μια ιστορική αναδρομή στα θέματα αυτά.
- Στο δεύτερο κεφάλαιο γίνεται μια πιο αναλυτική παρουσίαση της πλατφόρμας εργασίας CMU Sphinx καθώς και του Pocketsphinx παρουσιάζοντας τα γενικά χαρακτηριστικά τους, την δομή τους τον τρόπο λειτουργίας τους.
- Το τρίτο κεφάλαιο είναι αφιερωμένο αποκλειστικά στην περιγραφή του τρόπου δημιουργίας του Ελληνικού Ακουστικού Μοντέλου.
- Στο τέταρτο κεφάλαιο γίνεται παρουσίαση σεναρίων μελλοντικής χρήσης των αποτελεσμάτων της Μεταπτυχιακής διατριβής.
- Τέλος στο πέμπτο και τελευταίο κεφάλαιο καταγράφονται τα συμπεράσματα αλλά και οι επόμενες επιστημονικές αναζητήσεις του συγγραφέα.

2. Ορισμοί Βασικές έννοιες

ASR (Automatic Speech Recognition): Αυτόματη αναγνώριση φωνής.

CMU Sphinx : πλατφόρμα εργαλείων αυτόματης αναγνώρισης φωνής.

Pocketsphinx : πλατφόρμα αυτόματης αναγνώρισης φωνής προσαρμοσμένη για χρήση σε κινητά και φορητές συσκευές.

Arduino: πλατφόρμα μικρο επεξεργαστή ανοικτού κώδικα.

Raspberry Pi 3 : πλατφόρμα μικρο επεξεργαστή ανοικτού κώδικα.

Internet of things: η δικτύωση φυσικών συσκευών, οχημάτων, κτηρίων αλλά και άλλων αντικειμένων που τους επιτρέπει την συλλογή και ανταλλαγή δεδομένων μεταξύ τους.

Πετρογλυφικά: μορφή επικοινωνίας μέσω της χάραξης σχημάτων πάνω σε πέτρες.

Πικτογράμματα: μορφή επικοινωνίας μέσω της χάραξης συμβόλων που αναπαριστούν γεγονότα , αντικείμενα, δραστηριότητες.

Ιδεογράμματα: γραφιστικά σύμβολα που αντιπροσώπευαν κάποια ιδέα.

DARPA: Defense Advanced research Projects Agency Στρατιωτική υπηρεσία ερευνών και μελετών των ΗΠΑ.

ARPANET (Advanced Research Projects Agency Network): το πρώτο στον κόσμο δίκτυο μεταγωγής πακέτου και το δίκτυο πυρήνας ενός συνόλου που θα συνέθετε το παγκόσμιο Διαδίκτυο (internet).

TCP/IP (Transmission Control Protocol/Internet Protocol): συλλογή πρωτοκόλλων επικοινωνίας στα οποία βασίζεται το διαδίκτυο.

Gopher: διαδίκτυακό πρωτόκολλο επικοινωνίας του 1990.

Word-wide web: ανοιχτό σύστημα διασυνδεδεμένων πληροφοριών και πολυμεσικού περιεχομένου.

CERN (Conseil Européen pour la Recherche Nucléaire) : το μεγαλύτερο σε έκταση (πειραματικό) κέντρο πυρηνικών ερευνών και ειδικότερα επί της σωματιδιακής φυσικής στον κόσμο.

Λόγος: η διανοητική ικανότητα του ανθρώπου να σχηματίζει και να εκφράζει τις σκέψεις του με την ομιλία.

Recognition hypothesis: μια γραμμή κειμένου που προκύπτει σαν αποτέλεσμα της αναγνώρισης.

Dictionary (dic): λίστα λέξεων μιας γλώσσας.

Ακουστικό μοντέλο-Acoustic model: μια σειρά αρχείων που περιέχουν στατιστικές πληροφορίες σχετικά με βασικές μονάδες του λόγου.

Γλωσσικό μοντέλο-Language model: αρχείο που περιγράφει την γλώσσα που θέλουμε να αναγνωρίσουμε.

Cortana: προσωπικός ηλεκτρονικός βοηθός του λειτουργικού συστήματος windows.

Siri: προσωπικός ηλεκτρονικός βοηθός της Apple για τα κινητά τηλέφωνα iPhone.

Φωνήματα-phonemes: πεπερασμένες, διακριτές φωνητικές μονάδες, που χαρακτηρίζονται από ένα σύνολο ακουστικών ιδιοτήτων.

Δυναμική χρονική στρέβλωση- Dynamic time warping-DTW: αλγόριθμος που μετράει ομοιότητες ανάμεσα σε δύο αλληλουχιών που μπορεί να διαφέρουν στον χρόνο ή την ταχύτητα.

Κρυφά Μαρκοβιανά Μοντέλα-Hidden Markov Models-HMM: στατιστικές Μαρκοβιανές διαδικασίες στις οποίες οι καταστάσεις είναι κρυφές στον παρατηρητή.

Νευρωνικό δίκτυο: δίκτυο από απλούς υπολογιστικούς κόμβους διασυνδεδεμένους μεταξύ τους.

Βαθιά Νευρωνικά Δίκτυα- Deep Neural Network-DNN: τεχνητά νευρωνικά δίκτυα τα οποία έχουν παραπάνω από ένα «στρώματα» από κρυμμένες μονάδες μεταξύ των εισόδων και των εξόδων τους.

Φωνητικό λεξικό (Phonetic dictionary): αντιστοίχιση των λέξεων μας σε μια μορφή, κατανοητή από την πλατφόρμα η οποία περιέχει φωνήματα

Front end μπλόκ: το μπλοκ διεπαφής και παρουσίασης του CMU Sphinx

Αποκωδικοποιητής (decoder): μέρος του CMU Sphinx όπου γίνεται η αποκωδικοποίηση του σήματος και η αναγνώριση των λέξεων

Διαχειριστής αναζήτησης: μέρος του αποκωδικοποιητή που κατασκευάζει ένα δέντρο πιθανοτήτων για την καλύτερη υπόθεση

Γλωσσολόγος: δημιουργεί το γράφημα αναζήτησης που ο αποκωδικοποιητής χρησιμοποιεί κατά την διάρκεια της αναζήτησης των λέξεων

Ακουστικός βαθμολογητής: υπολογίζει την πιθανότητα της κατάστασης εξόδου ή τις τιμές πυκνότητας πιθανότητας για τις διάφορες καταστάσεις

FWDTREE: Αλγόριθμος αναζήτησης όπου έχουμε χρήση αλγόριθμου Viterbi και ενός στατικού δεντρικής μορφής λεξιλογίου

FWDFLAT: Αλγόριθμος αναζήτησης όπου έχουμε χρήση αλγόριθμου Viterbi και ενός επίπεδου λεξιλογίου

BESTPATH: Αλγόριθμος αναζήτησης όπου έχουμε εύρεση γραφήματος λέξεων

sphinxbase: πρόγραμμα βάσης που περιέχει τις βιβλιοθήκες του rocketsphinx

sphinxtrain: πρόγραμμα εκπαίδευσης rocketsphinx

CMUCLMTK: πακέτο προγραμμάτων Carnegie Mellon University

text2wfreq: λίστα με όλες τις λέξεις που έχουμε καθώς και με έναν αριθμό που δείχνει τις συνολικές εμφανίσεις της κάθε λέξης

wfreq2vocab: παίρνει σαν είσοδο το αρχείο που παράγει το text2wfreq και μας δίνει σαν έξοδο ένα λεξιλόγιο σε μορφή ASCII

text2idngram : Δέχεται σαν είσοδο το λεξιλόγιο και μια ροή κειμένου και δίνει σαν έξοδο μια λίστα με όλους τους αύξοντες αριθμούς n-grams αριθμών που αντιστοιχούν στις χαρτογραφήσεις των λέξεων

idngram2lm : Παίρνει την είσοδο από το text2idngram και μας δίνει το Γλωσσικό μοντέλο

Aegisub: πρόγραμμα επεξεργασίας υποτίτλων

binary: μορφή αρχείου που δεν είναι κείμενο

ARPA: Advanced Research Projects Agency Η υπηρεσία από όπου πρέκυνψε η DARPA

ARPABET: αλφάβητο που αναπτύχθηκε από την ARPABET

PHONETISAUROS: πρόγραμμα μετατροπής λέξεων σε σύνολο φωνημάτων

sequitur: πρόγραμμα μετατροπής λέξεων σε σύνολο φωνημάτων

an4: βάση εκπαίδευσης rocketsphinx

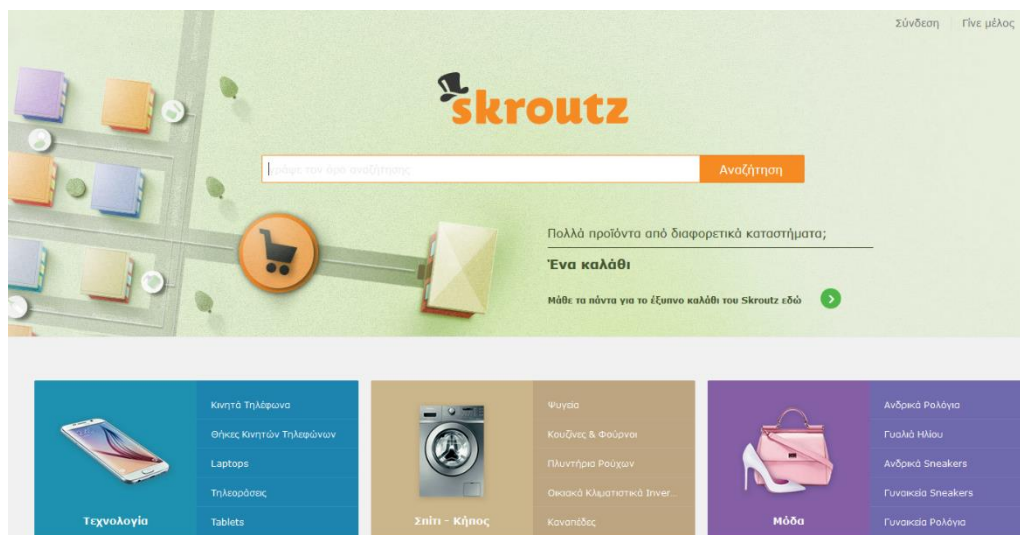
ROV (Remotely Operated Vehicles): Αυτόνομο υποβρύχιο ρομποτικό όχημα στο οποίο υπάρχει απομακρυσμένος χειρισμός με την βοήθεια καλωδίου διασύνδεσης

ΚΕΦΑΛΑΙΟ 1

Επικοινωνία Λόγος Αυτόματη Φωνητική Αναγνώριση

§1. Επικοινωνία. Από την αρχή της ιστορίας μέχρι σήμερα

Στην εποχή μας όλο και μεγαλύτερα μέρη του ενεργού πληθυσμού έρχονται σε καθημερινή επαφή με τις υπηρεσίες που προσφέρει το διαδίκτυο. Κάνοντας χρήση ενός μεγάλου αριθμού εφαρμογών πλέον δίνεται η δυνατότητα να φέρουμε σε πέρας εργασίες που υπό άλλες συνθήκες θα απαιτούσαν περισσότερο χρόνο και κόπο. (Εικόνα 3)



Εικόνα 3. Η πλατφόρμα συγκριτικής αναζήτησης skrutz

Ταυτόχρονα έχουμε την δυνατότητα να ενημερωθούμε, να ψάξουμε για θέματα που μας ενδιαφέρουν, να διαμοιράσουμε πληροφορία σε άλλες ομάδες πληθυσμού. Ουσιαστικά σήμερα το διαδίκτυο είναι ο τελευταίος κρίκος σε μια σειρά τεχνολογιών επικοινωνίας (Odlyzko, 2000), και μάλιστα είναι μια ραγδαία εξελισσόμενη τεχνολογία που αυτή την στιγμή οδηγείται προς την απλότητα και την δια δρόση διαφόρων επιμέρους εφαρμογών εμπλέκοντας ενεργά όλο και πιο πολύ τον απλό χρήστη. Στην εποχή μας το διαδίκτυο, σταθερά μεταλλάσσεται σε Internet of things (Rose, Eldridge, & Lyman, 2015), εξελισσόμενο και

αναβαθμιζόμενο εκμεταλλευόμενο την ραγδαία εξέλιξη των νέων τεχνολογιών αλλά και την μείωση των κοστολογίων στην χρήση και απόκτηση νέων υπολογιστικών συστημάτων.

Βέβαια αυτό που σήμερα είναι μια απλή κίνηση για πολλούς από εμάς, η αποστολή για παράδειγμα ενός email ή ενός μηνύματος, μια κίνηση που παίρνει δευτερόλεπτα για να ολοκληρωθεί, χρειάστηκε αιώνες εξέλιξης και ανακαλύψεων ώστε να κατορθώσει να είναι τόσο απλή πλέον, και μέρος της καθημερινότητας μας.

Μιλώντας για επικοινωνία φυσικά μιλάμε για μεταφορά πληροφορίας από το ένα μέρος σε κάποιο άλλο κάνοντας χρήση ενός μέσου. Η μεταφορά πληροφορίας μπορεί να είναι είτε ένα απλό «καλημέρα» σε κάποιο οικείο πρόσωπο σε μας, είτε η μεταφορά μεγάλου όγκου πληροφορίας σε πολλαπλούς αποδέκτες ταυτόχρονα (live video streaming). Η ανθρώπινη επικοινωνία πήρε άλλη διάσταση με την δημιουργία της γλώσσας περίπου 500000 χρόνια πριν, ενώ τα πρώτα σύμβολα, σχέδια τα οποία έχουμε σαν απόδειξη επικοινωνίας έχουν ηλικία 30000 χρόνια και έχουν ανακαλυφθεί στην σπηλιά του Chauvet στην Γαλλία. Μέσα από μια σειρά μοναδικών σχεδίων (Εικόνα 4) στους τοίχους της σπηλιάς ο ανώνυμος δημιουργός μας μεταφέρει έναν μεγάλο όγκο πληροφοριών σχετικά με τον καθημερινό τρόπο ζωής των τότε ανθρώπων.



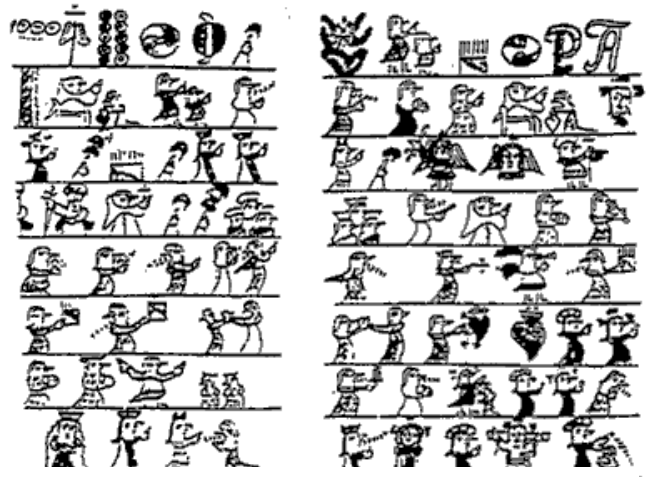
Εικόνα 4. Σχέδια από την σπηλιά του Chauvet
(Bradshaw Foundation)

Χρειάστηκαν περίπου 20000 χρόνια ώστε να περάσουμε, γύρω στο 10000 π.Χ. στην επόμενη μορφή επικοινωνίας που είναι τα πετρογλυφικά, χαράξεις δηλαδή σχημάτων πάνω σε πέτρες.

Ενδεχόμενα φυσικά να υπήρχαν και άλλες μορφές επικοινωνίας εκείνη την εποχή, αλλά αυτές έχουν χαθεί σαν καταγραφή και η μόνη μορφή που σήμερα γνωρίζουμε είναι τα πετρογλυφικά.

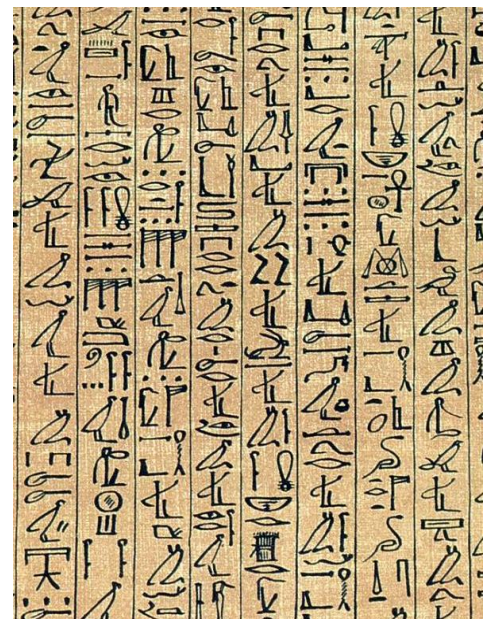
Η επόμενη εξέλιξη στην επικοινωνία ήταν η δημιουργία των πικτογραμμάτων. Πρόκειται για σχεδιαστικές αναπαραστάσεις (Εικόνα 5) που αναφέρονται σε δραστηριότητες, τόπους, δράσεις, ιστορίες και ουσιαστικά έχουμε την εξιστόρηση ενός γεγονότος, οπότε μπορούν να καταταχθούν και χρονολογικά.

Διάφοροι πολιτισμοί γύρω στο 9000 π.Χ. έκαναν χρήση πικτογραμμάτων τα οποία έγιναν ιδιαίτερα δημοφιλή το χρονικό διάστημα 6000-5000 π.Χ. αποτελώντας ταυτόχρονα και την βάση για ιερογλυφικά αλλά και τα ολογραφικά συστήματα γραφής που ακολούθησαν. Με την πάροδο του χρόνου τα πικτογράμματα εξελίχθηκαν σε ιδεογράμματα, γραφιστικά σύμβολα που αντιπροσώπευαν κάποια ιδέα. Λόγω της οικουμενικότητας των ιδεών πολλά ιδεογράμματα είναι κοινά για διάφορους πολιτισμούς, αποτελώντας ταυτόχρονα τους «προγόνους» των ολογραφικών συστημάτων γραφής όπως είναι τα Αιγυπτιακά ιερογλυφικά και η Κινέζικη γραφή.



Εικόνα 5. Πικτόγραμμα του 1510 που αναφέρεται στην έλευση μισθοφόρων στο νησί Ισπανιόλα (Wikipedia 2017)

Το επόμενο μεγάλο βήμα στην εξέλιξη της επικοινωνίας υπήρξε η χρήση της γραφής. Τα πρώτα συστήματα γραφής εμφανίστηκαν περί το 4000 π.Χ. και εξελίχθηκαν γύρω στο 3000 π.Χ. σε αποτυπώσεις πάνω σε πλάκες αργίλου (Σουμερίοι). Στις πρώτες αυτές φάσεις της γραφής, κυρίως γινόταν αποτύπωση εμπορικών διαδικασιών, απογραφών αλλά και συμφωνιών. Με την χρήση λογογραμμάτων, χαρακτήρων δηλαδή που αντιπροσώπευαν λέξεις ή φράσεις, γινόταν η καταγραφή όλων των παραπάνω ενώ γύρω στο 2600 π.Χ. είχαμε και την αντιπροσώπευση συλλαβών μέσω των λογογραμμάτων (Εικόνα 6). Τελικά το σύστημα γραφής των Σουμεριών πάνω στις πλάκες αργίλου, κατέληξε να χρησιμοποιείται γενικά και πέραν των εμπορικών συναλλαγών, περιεχοντας λογογράμματα, συλλαβές αλλά και νούμερα.



Εικόνα 6. Αιγυπτιακά ιερογλυφικά (Wikipedia 2017)

Στον Ελλαδικό χώρο την ίδια περίοδο έχουμε την χρήση της Μινωικής ιερογλυφικής γραφής (Εικόνα 7) από όπου προέκυψε η Γραμμική Α (Marangozis, 2007), γλώσσες που μέχρι και σήμερα δεν έχουν αποκρυπτογραφηθεί. Ουσιαστικά όμως η πρώτη γραφή της Ελληνικής γλώσσας είναι η Γραμμική Β που ονομάστηκε έτσι από τον Άρθουρ Έβανς λόγω της χρήσης γραμμικών στοιχείων στην

αναπαράσταση, έναντι της χρήσης εικονογραμμάτων. Είναι προφανές ότι η δημιουργία πλέον των Αλφαβήτων, τυποποιημένων δηλαδή μορφών απεικονίσεων απλών μονών συμβόλων που αντιστοιχούσαν σε απλά φωνήματα βοήθησε στην περαιτέρω εξέλιξη και αλλά και εξάπλωση της γραφής και συνεπώς της επικοινωνίας. Η γραφή δεν ήταν πια προνόμιο ορισμένων «ειλεκτών» κάθε κοινωνίας αλλά μπόρεσε να διαδοθεί και σε μια σειρά άλλων κοινωνικών στρωμάτων.



Εικόνα 7. Κρητικά ιερογλυφικά πάνω σε πράσινη σφραγίδα Ίασπη. 1800 π.Χ (Wikipedia 2017)

αλλάζουν τα μέσα της επικοινωνίας πάντα με γνώμονα την πιο γρήγορη, άμεση και ασφαλή μετάδοση της πληροφορίας. Μπορεί τα βήματα της ανθρωπότητας να ήταν στην αρχή αργά και επίπονα, αλλά στην εποχή μας οι εξελίξεις στα μέσα επικοινωνίας είναι πλέον καθημερινές πολύ γρήγορες και σε αρκετές περιπτώσεις δύσκολες να παρακολουθήσεις. Χρειάστηκε βέβαια πρώτα να περάσουμε από τα «νηπιακά» στάδια όπου η αποστολή μηνύματος γινόταν με την χρήση ειδικών απεσταλμένων (ανθρώπων ή ζώων), κάτι που προϋπόθετε το μήνυμα να είναι γραμμένο σε κάτι πιο εύχρηστο από τον πηλό. Η ακόμα και από το στάδιο όπου η μετάδοση μηνύματος γινόταν με την χρήση σημάτων καπνού, φωτιάς, τυμπάνων η κατόπτρων.

Η διαχρονική εξέλιξη των μέσων της επικοινωνίας, από εκεί και μετά θα μπορούσε να παρουσιαστεί πολύ αναλυτικά με την βοήθεια του σχεδιαγράμματος 1, όπου έχουν αποτυπωθεί οι κυριότεροι σταθμοί αυτής της εξέλιξης. Παρατηρεί κανείς ότι σημαντικές ανακαλύψεις



Εικόνα 8. Δέκτης σημάτων Morse (Wikipedia 2017)

Η εξέλιξη που είχαμε στις θετικές επιστήμες ταυτόχρονα οδηγούσε και σε νέες ανακαλύψεις και εφαρμογές στα μέσα επικοινωνίας. Οι ανακαλύψεις για παράδειγμα των πενών της τυπογραφίας των γραφομηχανών των τηλεγράφων των σημάτων Morse των ραδιοκυμάτων, των τηλεφώνων, του ραδιοφώνου της τηλεόρασης της δορυφορικής επικοινωνίας συνέτειναν στο να έχουμε κάθε φορά όλο και πιο εύκολη όλο και πιο άμεση μετάδοση πληροφορίας άρα και επικοινωνία.

Μέχρι τον 20^ο αιώνα

- 3000 π.Χ. Χρήση πάπυρων
- 3000 π.Χ. Χρήση τυμπάνων
- 2400 π.Χ. Πρώτα συστήματα ταχυδρομείου
- 59 π.Χ. Πρώτη εφημερίδα
- 1000 μ.Χ. χρήση πέννας
- 1400 μ.Χ. Πρώτα τυπογραφεία
- 1500 μ.Χ. Χρήση μολυβιών
- 16^{ος} αιώνας Χρήση σημαιών ναυτιλίας
- 1800 Χρήση γραφομηχανών
- 1838 Χρήση τηλεγράφου
- 1848 Ανακάλυψη τηλεφώνου
- 1896 Ανακάλυψη Ραδιοφώνου
- 1897 Ανακάλυψη ΗΥ
- 19^{ος} αιώνας χρήση λαμπών για σινιάλα

Μετά τον 20^ο αιώνα

- 1927 Ανακάλυψη τηλεόρασης
- Δεκαετία 1960 Δικτύωση ΗΥ
- 1983 Internet
- Δεκαετία 1980 Κινητά τηλέφωνα, Δίκτυα 1 G
- Δεκαετία 1990 Κινητά τηλέφωνα δίκτυα 2 & 2.5 G
- Μετά το 2000 Κινητά τηλέφωνα ,δίκτυα 3&4 G,διαδικτυακές υπηρεσίες,Facebook, skype, twitter, Instagram, YouTube ,blogger

Σχεδιάγραμμα 1. Η διαχρονική εξέλιξη στα μέσα επικοινωνίας

Αυτό που άλλαξε ριζικά τον τρόπο που σήμερα επικοινωνούμε και λειτουργούμε ξεκίνησε στα εργαστήρια του MIT (Leiner et al., 1997) όπου ο J.C.R. Licklider έθεσε το

θεωρητικό υπόβαθρο της ανταλλαγής «πακέτων» δεδομένων μεταξύ δικτυωμένων ηλεκτρονικών υπολογιστών. Το 1965 ο ερευνητής του MIT Lawrence G. Roberts κατάφερε να συνδέσει έναν υπολογιστή TX-2 στην Μασαχουσέτη με έναν υπολογιστή Q-32 στην Καλιφόρνια, μέσω μιας χαμηλής ταχύτητας κλήσης τηλεφωνικής γραμμής. Το πείραμα αυτό απέδειξε ότι η μετάδοση πακέτων δεδομένων ήταν εφικτή και το 1966 ο Roberts μετέβη στην DARPA με στόχο να αναπτύξει ένα δίκτυο υπολογιστών, παρουσιάζοντας το 1967 το ARPANET. Η γένεση συνεπώς αυτού που σήμερα ονομάζουμε Word-Wide Web είχε καθαρά στρατιωτική υποστήριξη και χρηματοδότηση ιδίως στα πρώτα δύσκολα βήματα του.

Μέχρι και το 1983 η χρήση του ARPANET γινόταν αποκλειστικά από στρατιωτικούς οργανισμούς και οργανισμούς έρευνας και ανάπτυξης. Το 1983 με την εισαγωγή του TCP/IP έγινε δυνατόν να διαχωριστεί η χρήση σε καθαρά στρατιωτική (Milnet) και σε καθαρά ερευνητική (ARPA Internet). Μέχρι το 1985 υπήρχαν περίπου 2000 «οικοδεσπότες» του δικτύου (Campbell-Kelly, Garcia-Swartz, & Campbell-Kelly, 2013), παρότι βέβαια ο αριθμός των δικτυωμένων παγκοσμίως υπολογιστών ήταν πολύ μεγαλύτερος.

Όπως αναφέρθηκε και παραπάνω η εξέλιξη των μέσων επικοινωνίας είναι άρρηκτα συνδεδεμένη με τις εξελίξεις στις θετικές επιστήμες και δη στις επιστήμες που σχετίζονται με τα



Εικόνα 9. Υπολογιστής σε σπίτι το 1965
(Mary Allen Wilkes 2017)

υπολογιστικά συστήματα. Ενώ μέχρι την δεκαετία του 1980 τα υπολογιστικά συστήματα ήταν ογκώδη (Εικόνα 9) και προσβάσιμα μόνον από ορισμένο αριθμό ατόμων (κατά κύριο λόγο ερευνητές η στρατιωτικοί) κατά την διάρκεια της δεκαετίας 1980 είχαμε την εμφάνιση των πρώτων προσωπικών υπολογιστών (Εικόνα 10). Μικρότεροι σε μέγεθος, οικονομικά μη απαγορευτικοί άρχισαν σιγά σιγά να «εισβάλουν» στα σπίτια μας, στην αρχή εκτελώντας απλές εργασίες υπολογιστικού χαρακτήρα η γραφής, ενώ αποτέλεσαν και την πρώτη βάση ανάπτυξης για τα πρώτα παιχνίδια διασκέδασης.

Ξαφνικά με την εμφάνιση των προσωπικών υπολογιστών υπήρξε ένα τεράστιο καταναλωτικό κοινό για υπηρεσίες online και παρότι αυτή η προοπτική αγνοήθηκε επιδεικτικά από τους κατασκευαστές προσωπικών υπολογιστών

ερασιτεχνικές προσπάθειες κατάφεραν να δημιουργήσουν τις πρώτες online κοινότητες, όπου υπήρχε ανταλλαγή μηνυμάτων (email) αλλά και οι πρώτες εμπορικές συναλλαγές. Η ραγδαία

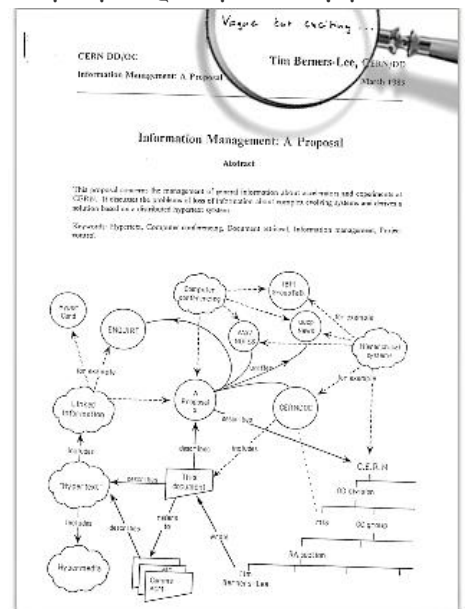
αύξηση των χρηστών του Internet (βασιζόμενο στο πρωτόκολλο TCP/IP), οδήγησε κατά την διάρκεια της δεκαετίας του 1990 στην εμφάνιση και των πρώτων περιηγητών (browsers).

Θα πρέπει όμως να δούμε σε ποια φάση της εξέλιξης στα μέσα επικοινωνίας βρισκόμαστε αυτή την εποχή. Οι δεκαετίες του 1980 και 1990 χαρακτηρίστηκαν από την εμφάνιση και «άνθησή» της ιδιωτικής τηλεόρασης και ραδιοφώνου. Υιοθετώντας ένα πολύ , οικείο στον τηλεθεατή και ακροατή στυλ, ιδίως η ιδιωτική τηλεόραση εισβάλε στα σαλόνια των σπιτιών μας και μας «κατακτά», αφού προβάλλει θέματα που μας αφορούν άμεσα. Ουσιαστικά η τηλεόραση δεν αφήνει και πολλά περιθώρια ανταγωνισμού στο Internet. Η «έγχρωμη» της εικόνα, η θεματολογία της η αμεσότητα της την κάνουν το κυρίαρχο μέσω επικοινωνίας της εποχής. Πόλεμοι μεταδίδονται πια σε ζωντανή σύνδεση, θέματα του απλού κόσμου βρίσκουν διέξοδο επικοινωνιακή στην τηλεόραση. Την ίδια περίοδο ο (αρχές της δεκαετίας του 1990) το πρωτόκολλο επικοινωνίας gopher ,εμφανίζεται και γίνεται σιγά σιγά διάσημο. Το πρωτόκολλο ήταν προσανατολισμένο προς τον σχεδιασμό που διασύνδεε το μενού επιλογών μας με συγκεκριμένα αρχεία.

Η εφεύρεση του World Wide Web από τον Tim Berners-Lee (Berners-Lee, Cailliau, Luotonen, Frystyk Nielsen, & Secret, 1994), την περίοδο που εργαζόταν στο CERN, είναι ουσιαστικά αυτό που πλέον ριζικά άλλαξε τα πάντα στο θέμα της επικοινωνίας και της διασυνδεδεμένης αλληλεπίδρασης. Η σημαντική διαφοροποίηση που το www εισήγαγε, ήταν ότι απλά από ένα έγγραφο κάποιος εύκολα μπορούσε να φτάσει σε κάποιο άλλο, καθώς το αρχικό έγγραφο εμπεριείχε διασύνδεση προς το έγγραφο προορισμό. Ο τρόπος διαμοίρασης και χρήσης της πληροφορίας που πρότεινε ο Berners-Lee διέφερε ριζικά από την φιλοσοφία του gopher και κατά την εξέλιξη της δεκαετίας το 1990 το www κυριάρχησε σαν πρωτόκολλο και τρόπος διαμοίρασης πληροφορίας κυριάρχησε. Προς το τέλος της δεκαετίας είχαμε πιο φιλικούς περιηγητές πιο εύχρηστα προγράμματα για αποστολή



Εικόνα 10. Προσωπικός υπολογιστής της δεκαετίας του 1980 (Wikipedia 2017)



Εικόνα 11. Η αρχική πρόταση για το www
(<https://www.home.cern>)

email και μια σειρά άλλων εφαρμογών όπως το irc η το icq που επέτρεπαν την online ταυτόχρονη αποστολή μηνύματος σε άλλους χρήστες.

Την ίδια χρονική περίοδο είχαμε την ανάπτυξη των δικτύων κινητής τηλεφωνίας που επέτρεψαν την αποστολή μηνυμάτων αποτελώντας ακόμα μια εξέλιξη που χαρακτήρισε τα μέσα επικοινωνίας της εποχής.

Η περεταίρω εξέλιξεις στον τομέα των μικρό επεξεργαστών αλλά και γλωσσών προγραμματισμού οδήγησαν μετά το 2000 στην «έκρηξη» των μέσων επικοινωνίας, που ζούμε σήμερα. Μια σειρά από πρωτοποριακές εφαρμογές εμφανίζονται και αλλάζουν για πάντα τον τρόπο με τον οποίο επικοινωνούμε όχι απλά μεταξύ μας, αλλά και με μια σειρά μηχανών, εντεταλμένων να μας «υπηρετούν». Τα κινητά τηλέφωνα αρχίζουν σιγά σιγά να ανταγωνίζονται τους προσωπικούς υπολογιστές ενώ όλο και πιο εξελιγμένα πρωτόκολλα επικοινωνίας δρομολογούνται ώστε να καλύψουν τις όλο και πιο μεγάλες απαιτήσεις των χρηστών του Internet. Εφαρμογές όπως το [yahoo](#), το [skype](#), το [twitter](#), το [Instagram](#), το [Youtube](#), το [facebook](#) με εκατομμύρια χρήστες σε όλο τον κόσμο (Εικόνα 12) έχουν πλέον αναγάγει το θέμα της επικοινωνίας σε ζήτημα δευτερολέπτων. Η διασύνδεση συσκευών, κτηρίων, υποδομών οδήγησε στο Internet of things τοποθετώντας τον άνθρωπο και τις ανάγκες του στο επίκεντρο της επικοινωνίας.

WORLD INTERNET USAGE AND POPULATION STATISTICS MARCH 25, 2017 - Update						
World Regions	Population (2017 Est.)	Population % of World	Internet Users 31 Mar 2017	Penetration Rate (% Pop.)	Growth 2000-2017	Users % Table
Africa	1,246,504,865	16.6 %	345,676,501	27.7 %	7,557.2%	9.3 %
Asia	4,148,177,672	55.2 %	1,873,856,654	45.2 %	1,539.4%	50.2 %
Europe	822,710,362	10.9 %	636,971,824	77.4 %	506.1%	17.1 %
Latin America / Caribbean	647,604,645	8.6 %	385,919,382	59.6 %	2,035.8%	10.3 %
Middle East	250,327,574	3.3 %	141,931,765	56.7 %	4,220.9%	3.8 %
North America	363,224,006	4.8 %	320,068,243	88.1 %	196.1%	8.6 %
Oceania / Australia	40,479,846	0.5 %	27,549,054	68.1 %	261.5%	0.7 %
WORLD TOTAL	7,519,028,970	100.0 %	3,731,973,423	49.6 %	933.8%	100.0 %

Εικόνα 12. Χρήστες υπηρεσιών Internet Μάρτιος 2017 (<https://www.internetworldstats.com>)

Σίγουρα είναι πολύ δύσκολο να περιγράψει κανείς αναλυτικά όλα τα διαφορετικά μέσα που σήμερα χρησιμοποιούμε για να επικοινωνήσουμε μεταξύ μας. Άλλωστε δεν είναι και ο σκοπός της διατριβής αυτός. Είναι δεδομένο ότι συνεχώς έχουμε την ανάπτυξη εφαρμογών που θεωρητικά κάνουν την ζωή μας πιο εύκολη και πιο ευχάριστη, προσφέροντας μας ευκολίες που άλλοτε δεν υπήρχαν. Με απλές διαδικασίες μπορούμε να πληρώσουμε πια τους

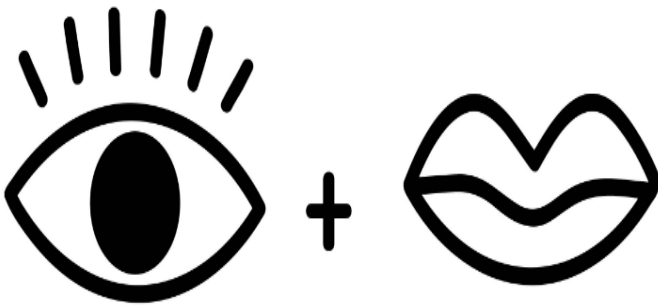
λογαριασμούς μας να επικοινωνήσουμε οπτικά με αγαπημένα πρόσωπα, να συμμετάσχουμε σε απομακρυσμένες επαγγελματικές συσκέψεις.

Η ιδέα του ελέγχου των διάφορων μηχανικών κατασκευών μας, με απλό τρόπο, φυσικά δεν είναι τωρινή. Ο πιο άμεσος απλός τρόπος επικοινωνίας μεταξύ ανθρώπων τουλάχιστον ήταν και είναι μέχρι στιγμής ο λόγος. Πρώτιστα ο προφορικός και μετά ο γραπτός.

Στην επόμενη ενότητα θα αναφερθούμε στον λόγο. Προφορικό και γραπτό. Στην εφαρμογή μας άλλωστε, κάνουμε χρήση και των δυο μορφών του λόγου, μετατρέποντας στην ουσία τον προφορικό σε γραπτό, οπότε είναι σκόπιμο να αναφέρουμε τα πιο κύρια χαρακτηριστικά του λόγου.

§2. Προφορικός και Γραπτός λόγος

Ο προφορικός και ο γραπτός λόγος (Εικόνα 13) αποτελούν δύο ισότιμες μορφές επικοινωνίας και ταυτόχρονης μετάδοσης πληροφορίας, με ιδιαίτερα ωστόσο χαρακτηριστικά που οριοθετούν επί της ουσίας τα πλεονεκτήματα και τα μειονεκτήματά τους.



Εικόνα 13. Γραπτός και προφορικός λόγος
(<https://www.mcad.edu>)

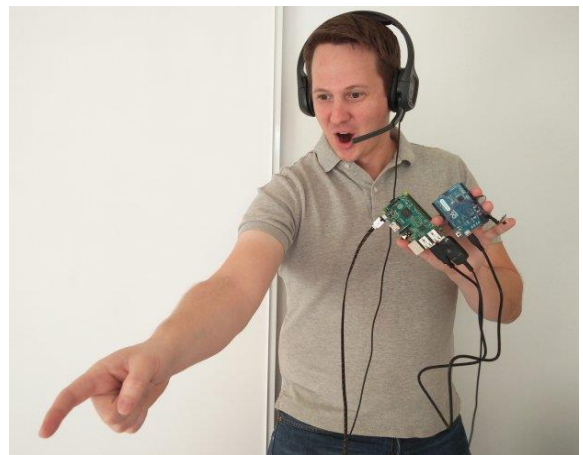
λόγου.

Είναι σωστό να αναφέρουμε στο σημείο αυτό τα διάφορα πλεονεκτήματα και μειονεκτήματα που έχουν οι δύο αυτές μορφές της επικοινωνίας και ταυτόχρονα να τις αναγάγουμε στο πεδίο των δικών μας ενδεχομένως εφαρμογών. Ο προφορικός λόγος ως γνωστόν χαρακτηρίζεται από αυθορμητισμό και ζωντάνια. Ένα σημαντικό πλεονέκτημα έναντι του γραπτού λόγου που όμως θα πρέπει να ελέγξουμε αν θελήσουμε να αναπτύξουμε εφαρμογές. Σε εφαρμογές ελέγχου μηχανικών κατασκευών ο προφορικός μας λόγος θα πρέπει να είναι σαφής, μη αυθόρμητος αλλά να παραμένει σίγουρα ζωντανός.

Είναι γνωστόν ότι ο προφορικός λόγος ικανοποιεί τις καθημερινές μας ανάγκες και δη τις τρέχουσες ανάγκες μας αποτελώντας το άμεσο εργαλείο επικοινωνίας μας με τον γύρω κόσμο.

Και όχι μόνον. Με την χρήση του προφορικού λόγου (Εικόνα 14) μπορούμε ελέγχοντας μια σειρά

Ένας γενικός ορισμός για τον λόγο θα ήταν, ότι είναι η διανοητική ικανότητα που έχει ο άνθρωπος να εκφράζει τις σκέψεις του και τα συναισθήματά του με την ομιλία. Όπως αναφέραμε και στην προηγούμενη ενότητα στην αρχή είχαμε την μετάδοση της πληροφορίας μεταξύ των ανθρώπων, μέσω του προφορικού λόγου ενώ στην συνέχεια με την αποτύπωση της πληροφορίας μέσω των διαφόρων σχεδίων είχαμε τις απαρχές του γραπτού



Εικόνα 14. Έλεγχος συσκευών με την χρήση
προφορικού λόγου
(<https://www.makezine.com>)

αισθητήρων για παράδειγμα να ελέγχουμε τον φωτισμό του σπιτιού μας, η την θέρμανση του ή το ήχο σύστημα μας. Η διαρκής εξέλιξη της τεχνολογίας, μας προσφέρει πλέον πολλές λύσεις που έχουν σχεδιαστεί με γνώμονα την καλυτέρευση της καθημερινής μας ζωής, και που έχουν σαν πυρήνα ανάπτυξης την χρήση του προφορικού λόγου. (“Use Raspberry Pi for Voice Control of Your Roomba | Make;,” 2017)

Η μετάδοση των μηνυμάτων στον προφορικό λόγο γίνεται με αμεσότητα και φυσικότητα και αυτό είναι ένα χαρακτηριστικό του που πρέπει να λάβουμε σοβαρά υπόψη μας κατά την ανάπτυξη των εφαρμογών μας.

Ο προφορικός λόγος είναι σαφώς πιο παραστατικός και προσφέρει στον ομιλητή τη δυνατότητα να ενισχύει τη νοηματική αξία των λέξεων του με τη χρήση παραγλωσσικών στοιχείων, όπως είναι οι εκφράσεις του προσώπου και οι χειρονομίες. Είναι ιδιαίτερο προνόμιο του προφορικού λόγου πως με τον επιτονισμό και μόνο της φωνής μια απλή λέξη ή έκφραση μπορεί να λάβει πλήθος νοηματικών προεκτάσεων, πλουτίζοντας έτσι το λόγο με τρόπους που δεν είναι εφικτοί στο γραπτό κείμενο. Όλο το φάσμα των συναισθημάτων μπορούν να περάσουν στην ομιλία του ατόμου ακριβώς μέσω του επιτονισμού, χωρίς να απαιτείται η ρητή λεκτική δήλωσή τους. Το συγκεκριμένο χαρακτηριστικό του προφορικού λόγου θα πρέπει να ληφθεί υπόψη, κατά τον σχεδιασμό εφαρμογών, καθώς αυτό που έχει σημασία είναι η σωστή λεκτική αποτύπωση του εκφερόμενου λόγου ασχέτως των διαφόρων συναισθηματικών ή άλλων τονισμών. Η παραστατικότητα του προφορικού λόγου θα πρέπει να μεταφερθεί επακριβώς σε γραπτό λόγο ανεξαρτήτως της συναισθηματικής φόρτισης του.

Ο προφορικός λόγος είναι προνόμιο όλων ανεξαρτήτως ηλικίας ή μορφωτικού επιπέδου, χαρακτηριστικό που τον διαφοροποιεί σημαντικά σε σχέση με τον γραπτό λόγο. Είναι δηλαδή εφικτό να μπορέσουμε να προσδιορίσουμε ένα αριθμό ελαχίστων λέξεων που να ικανοποιούν το λεξιλόγιο μιας γλώσσας και στην συνέχεια να αναπτύξουμε εφαρμογές που να καλύπτουν όσο τον δυνατόν μεγαλύτερα ποσοστά πληθυσμού. Αυτό γίνεται με στατιστική μελέτη και επεξεργασία μεγάλων αρχείων προφορικού λόγου της συγκεκριμένης γλώσσας.

Είναι απλός και προφανώς συνιστά μορφή φυσικής έκφρασης και γι’ αυτό αποτελεί κτήμα όλων των ανθρώπων είτε έχουν παρακολούθησει συστηματικές σπουδές είτε όχι. Είναι έτσι πολύ σημαντικό να κατανοήσουμε ότι ο χρήστης της οποιαδήποτε εφαρμογής αναπτύξουμε, δεν θα χρειάζεται ειδική εκπαίδευση σε ειδικά λεξιλόγια και ορολογία για να χρησιμοποιήσει την εφαρμογή μας.

Οι διαπροσωπικές σχέσεις μεταξύ των ανθρώπων κατά κύριο λόγο βασίζονται στην επικοινωνία μέσω του προφορικού λόγου. Αυτό γιατί ο προφορικός λόγος ευνοεί την φιλική προσέγγιση, την συνεννόηση, την ανταλλαγή απόψεων, την έκφραση γνώμης, και άλλες μορφές

των ανθρώπινων διαπροσωπικών ενεργειών. Αυτό το χαρακτηριστικό του προφορικού λόγου, η φιλική προσέγγιση και συμπεριφορά, τείνει να συμπεριληφθεί και σε όλες τις εφαρμογές που πλέον αναπτύσσονται και ρυθμίζουν την δια δράση μεταξύ ανθρώπου και μηχανής. (Park, Lee, Kim, & Bien, 2008)

Ο προφορικός λόγος εξελίσσεται συνεχώς καθώς εκφράζει πάντα τις ανθρώπινες ανάγκες και αντιλήψεις καλλιεργώντας πάντα τον προβληματισμό την κριτική σκέψη και την αμφισβήτηση. Υπάρχει βέβαια το ενδεχόμενο να έχουμε ασάφειες λόγω των ομόηχων λέξεων, αλλά αυτό μπορεί να διορθωθεί άμεσα με διευκρινήσεις. Για παράδειγμα οι λέξεις «πιο», «ποιο» ηχούν ακριβώς το ίδιο έχουν όμως εντελώς διαφορετική έννοια που εξαρτάται από τις υπόλοιπες λέξεις της έκφρασης μας. Το συγκεκριμένο όμως χαρακτηριστικό, των ομόηχων λέξεων, θα πρέπει να προσεχθεί ιδιαίτερα όταν μιλάμε για ανάπτυξη εφαρμογών, όπου χρειαζόμαστε σαφήνεια και ξεκάθαρος ορισμός εννοιών.

Προφανώς στον προφορικό λόγο ,ο ομιλητής έχει την δυνατότητα να αλλάζει και να βελτιώνει συνεχώς τα λεγόμενα του και αυτό οδηγεί στην καλύτερη κατανόηση. Πλην όμως αυτό αποτελεί και μειονέκτημα όταν μιλάμε για εφαρμογές γιατί όπως έχει ήδη αναφερθεί χρειαζόμαστε σαφήνεια και άμεση κατανόηση του μεγαλύτερου ποσοστού των λεγομένων.

Ένα γενικό μειονέκτημα του προφορικού λόγου είναι το γεγονός ότι είναι εφήμερος. Αν δεν καταγραφεί κάπου, χάνεται. Συνεπώς χρειαζόμαστε μια σειρά από συσκευές (μικρόφωνο, ηλεκτρονικός υπολογιστής ελέγχου, κινητή συσκευή κτλ.) και λογισμικό (Samah & Osman, 2015) ώστε να μπορέσουμε να αντιπαλέψουμε το πρόβλημα αυτό και να έχουμε καταγεγραμμένο τον προφορικό λόγο σε μια μορφή που να εξυπηρετεί τον σχεδιασμό των εφαρμογών μας.



Εικόνα 15. Καταγραφή προφορικού λόγου σε κινητή συσκευή

(<https://www.nytimes.com>)

Επίσης χαρακτηρίζεται γενικά από προχειρότητα στην έκφραση, ελλειπτικότητα, επαναλήψεις, παύσεις, κενά, ασυνταξίες καθώς είναι αυτοσχεδιαστος και διαφορετικός για κάθε ομιλητή. Η έλλειψη τυποποίησης θέτει τα δικά της εμπόδια στις διάφορες αναπτύξεις εφαρμογών. Είναι βέβαια πιο αναλυτικός αλλά και λιγότερο ακριβής ενώ τον χαρακτηρίζει μια γενικευμένη ασάφεια. Πόσες φορές στην ερώτηση «τι κάνεις;» δεν έχουμε απαντήσει «καλά, ευχαριστώ», ενώ αν θέλαμε να είμαστε σωστοί στην

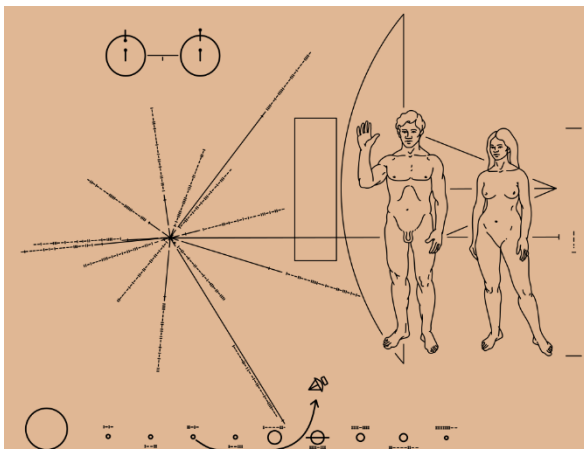
κυριολεξία θα έπρεπε να απαντήσουμε για το ποια ενέργεια κάνουμε ,για παράδειγμα, «γράφω κάτι στον υπολογιστή».

Τέλος ένα μειονέκτημα της επικοινωνίας μέσω του προφορικού λόγου είναι η μη ύπαρξη επαρκούς επεξεργασμένου λεξιλογίου με αποτέλεσμα την μη σωστή μεταφορά και πρόσληψη του μηνύματος από τον δέκτη στον πομπό.

Ο γραπτός λόγος που είναι χρονικά μεταγενέστερος του προφορικού έχει και αυτός τα δικά του πλεονεκτήματα αλλά και περιορισμούς. Καταρχήν είναι προσχεδιασμένος οπότε επιτρέπει τον έλεγχο και συνεπώς έχει επιμελημένη σύνταξη διατύπωση και φροντισμένο λεξιλόγιο. Μας δίνει την δυνατότητα να επεξεργαστούμε με προσοχή τις διατυπώσεις και τα εκφραστικά του μέσα



Εικόνα 16. Ο λίθος της Ροζέττας με την επιγραφή του σε τρεις διαφορετικές Γλώσσες (British Museum 2017)



Εικόνα 17. Γραφική παράσταση που φέρει ο Pioneer (By Vectors by Oona Räisänen (Mysid); designed by Carl Sagan & Frank Drake; artwork by Linda Salzman Sagan - Vectorized in CorelDRAW from NASA image GPN-2000-001623, Public Domain, <https://goo.gl/1QNB RN>)

επιτυγχάνοντας έτσι υψηλή ποιότητα λόγου. Απουσιάζουν άρα, από το γραπτό λόγο οι αστοχίες και οι προχειρότητες του προφορικού, γεγονός που τον καθιστά ιδανικότερο για τις πιο επίσημες αλλά και τεχνικές μορφές επικοινωνίας.

Ο γραπτός λόγος προσφέρει το αναγκαίο περιθώριο χρόνου προκειμένου να οργανώσουμε κατάλληλα τις ιδέες και τα επιχειρήματά μας. Οργάνωση που προσδίδει τελικά μεγαλύτερη ευστοχία και ακρίβεια στη γραπτή έκφραση σε σχέση με την αντίστοιχη προφορική. Το γραπτό κείμενο, επομένως, αποδίδει με μεγαλύτερη αποτελεσματικότητα και πληρότητα τις σκέψεις μας, εφόσον αυτές είναι οργανωμένες, επαρκώς

ανεπτυγμένες και έχουν επανελεγχθεί. Ταυτόχρονα η τα γραπτά κείμενα μας βοηθούν να κατανοήσουμε καλύτερα πολιτισμούς (Εικόνα 16) αλλά και να μελετήσουμε συγκριτικά. Στον γραπτό λόγο, εφόσον δεν έχουμε την δυνατότητα διευκρινήσεων, αναγκάζομαστε να έχουμε μια αυστηρά δομημένη ανάπτυξη των σκέψεων μας. Άλλωστε αποσκοπούμε και στην ορθή μετάδοση του μηνύματος μας προς τον δέκτη. Επιλέγοντας τις κατάλληλες λέξεις μπορούμε να αποδώσουμε το νόημα που εμείς θέλουμε σε κάθε έκφραση του. Η επιλογή κατάλληλου λεξιλογίου και συντακτικού μας επιτρέπει να κάθε φορά να δώσουμε στον λήπτη του μηνύματος, ακριβώς την ουσία των σκέψεων και συναισθημάτων μας, όπως αυτά έχουν αποτυπωθεί πάνω στο μέσο του γραπτού λόγου. Ένας προικισμένος συγγραφέας μπορεί με σαφήνεια να αποδώσει μέσα από μια σειρά γραμμών όλη την ουσία του μηνύματος που θέλει να μεταφέρει.

Ο γραπτός λόγος, το γραπτό μήνυμα είναι κάτι που διαρκεί στον χρόνο, σε σχέση με τον προφορικό λόγο. Με δεδομένο αυτό έχουμε την δυνατότητα σήμερα όχι μόνον να μελετάμε τα μηνύματα του παρελθόντος αλλά και να στέλνουμε και μηνύματα προς το μέλλον (Sagan, Sagan, & Drake, 1972) σαν μέρος της αέναης προσπάθειας μας για εξερεύνηση και γνώση (Εικόνα 17). Αυτός είναι και ο λόγος, για τον οποίο έχουμε μέχρι σήμερα καταφέρει να διασώσουμε την πολιτιστική κληρονομιά του ανθρώπου. Η ύπαρξη του γραπτού λόγου διευκολύνει την ανάπτυξη των γραμμάτων, των τεχνών και των επιστημών ωθώντας το άνθρωπο σε νέους δρόμους ανακάλυψης και γνώσης.

Σαφέστατα υπάρχουν και μειονεκτήματα αυτής της μορφής επικοινωνίας. Ο γραπτός λόγος μπορεί να χρησιμοποιηθεί μόνον από μορφωμένα άτομα, και όταν μιλάμε για γραπτό λόγο εφαρμογών, μιλάμε για απόλυτα μορφωμένα και εξιδεικευμένα άτομα.

Δεν χαρακτηρίζεται από αυθορμητισμό ενώ τις πιο πολλές φορές ακολουθεί μια τυπικότητα που συνήθως τον χαρακτηρίζει. Μη ακολουθώντας τις εξελίξεις της γλώσσας δεν ανταποκρίνεται πάντα στις σύγχρονες ανάγκες επικοινωνίας. Το να εκφράσουμε τις σκέψεις μας, τα συναισθήματα μας με την βοήθεια του γραπτού λόγου, απαιτεί χρόνο αλλά και λογική διεργασία από μέρους μας. Στον γραπτό λόγο συνήθως στερούμαστε την ανατροφοδότηση (feedback) καθώς δεν έχουμε την άμεση επικοινωνία του προφορικού λόγου. Ένα από τα κύρια μειονεκτήματα του γραπτού λόγου είναι ότι μπορεί να είναι πολυσήμαντος και αυτό να μας δημιουργήσει προβλήματα στην επικοινωνία καθώς δεν έχουμε την δυνατότητα της άμεσης παροχής επιπλέον διευκρινήσεων.

Στην συνέχεια θα αναφερθούμε στην αυτόματη φωνητική αναγνώριση ολοκληρώνοντας το πρώτο και κατά κάποιο λόγο «εισαγωγικό» κεφάλαιο της Μεταπτυχιακής διατριβής.

§3. Αυτόματη Φωνητική Αναγνώριση

3.1 Γενικά

Με τον όρο Αυτόματη Φωνητική Αναγνώριση –Automatic Speech Recognition (ASR) εννοούμε την διαδικασία μετατροπής ενός ηχητικού σήματος φωνής σε μια διαδοχική σειρά λέξεων με την χρήση κάποιου υπολογιστικού αλγόριθμου (Anusuya & Katti, 2009). Αποτελεί πεδίο ενεργούς έρευνας για τα τελευταία 60 τουλάχιστον χρόνια και σαν κύριος στόχος της ερευνητικής κοινότητας παραμένει η βελτίωση της ακρίβειας της αυτόματης φωνητικής αναγνώρισης για διάφορα είδη προφορικού λόγου για διάφορα περιβάλλοντα και ομιλητές. Ουσιαστικά με το ASR επιδιώκουμε η δια-δράση ανάμεσα σε άνθρωπο και μηχανή να είναι όσο το δυνατόν πιο ακριβής και εύχρηστη. Στο παρελθόν βέβαια η δια δράση αυτή δεν ήταν προτιμητέα. '

Η έλλειψη ισχυρών υπολογιστικών συστημάτων δεν μας επέτρεπε να έχουμε την απαιτούμενη ακρίβεια στις διαδικασίες ASR ενώ ταυτόχρονα η απόδοση των μηχανών αυτών δεν μπορούσε να υπερνικήσει άλλα πιο κλασικά συστήματα δια δράσης μας με τις μηχανές , όπως το πληκτρολόγιο και το ποντίκι. Αυτό βέβαια έχει αλλάξει την τελευταία δεκαετία (Yu, 2014) καθώς οι εξελίξεις στο τομέα των επεξεργαστών υπολογιστικών συστημάτων μας έχει επιτρέψει να έχουμε υπολογιστικά συστήματα με πολλαπλούς πυρήνες επεξεργασίας ενώ ταυτόχρονα μπορούμε να διασυνδέσουμε μια σειρά υπολογιστικών συστημάτων ώστε να εκτελούν την ίδια εργασία αυξάνοντας την επεξεργαστική ισχύ του συστήματος μας. Έτσι η διαδικασία της «εκπαίδευσης» που πρέπει να κάνουμε ώστε να έχουμε ένα ακριβές μοντέλο ASR μπορεί να προχωρήσει πιο γρήγορα και πιο εύκολα αποδίδοντας μας ένα τελικό αποτέλεσμα που έχει μεγαλύτερη ακρίβεια κατά την διαδικασία της ASR. Λόγω της ραγδαίας ανάπτυξης του Internet και των υπηρεσιών που αυτό μας προσφέρει, είναι πλέον εύκολο να έχουμε πρόσβαση σε μεγάλο όγκο πραγματικών δεδομένων που μπορούμε να χρησιμοποιήσουμε κατά την φάση της εκπαίδευσης του ASR μοντέλου μας. Μεγάλη ποσότητα πραγματικών δεδομένων μας επιτρέπει να έχουμε λιγότερες παραδοχές για το μοντέλο μας, και έτσι να οδηγηθούμε σε ένα μοντέλο ακριβές και σταθερό. Η μεθοδολογία αυτή χρησιμοποιήθηκε και στις εργασίες αυτής της Μεταπτυχιακής διατριβής, όπως και θα παρουσιαστεί αναλυτικά στο αντίστοιχο Κεφάλαιο 3.

Στην εποχή μας οι φορητές συσκευές, οι μεταφερόμενες συσκευές, οι έξυπνες συσκευές έγιναν πολύ δημοφιλείς, εύκολες στην χρήση και οικονομικά προσιτές προς όλους τους χρήστες. Σε όλες τις παραπάνω συσκευές η χρήση του πληκτρολογίου προφανώς παρουσιάζει κάποιες δυσχέρειες όποτε ήταν λογικό να αναζητηθούν λύσεις στην κατεύθυνση της δια δράσης

ανθρώπου μηχανής που να βασίζονται στον πιο απλό, παλιό και γνώριμο τρόπο επικοινωνίας του ανθρώπου, τον προφορικό λόγο.

3.2 Εφαρμογές

Τα τελευταία χρόνια έχουμε μια σειρά εφαρμογών στις οποίες η τεχνολογίες λόγου παίζουν έναν σημαντικό λόγο. Αυτές μπορούν να διαχωριστούν σε αυτές που βελτιώνουν την επικοινωνία μεταξύ ανθρώπων και σε αυτές που βελτιώνουν την επικοινωνία μεταξύ ανθρώπου και μηχανής.

Σαν παραδείγματα των πρώτων μπορούμε να αναφέρουμε τις εφαρμογές speech2speech που μας επιτρέπουν την αυτόματη μετάφραση των λεγομένων μας σε μια νέα γλώσσα (Stallard et al., 2009), βοηθώντας μας ουσιαστικά να καταργήσουμε λεκτικά εμπόδια που είχαμε και που μας εμπόδιζαν σε μια σειρά δραστηριοτήτων. Η παραπάνω τεχνολογία έχει ενσωματωθεί και σε δημοφιλείς και καθημερινές εφαρμογές όπως το skype (“Skype Translator Introduces Speech-to-Speech Translation in French and German – Translator,” 2017.) επιτρέποντας σε άτομα που μιλούν διαφορετικές γλώσσες να επικοινωνούν πλέον μεταξύ τους πιο εύκολα.

Την ίδια στιγμή οι εφαρμογές που προέρχονται από τις τεχνολογίες φωνής έχουν σημαντικά βελτιώσει την δια δράση μας με διάφορες μηχανικές κατασκευές. Οι πιο δημοφιλείς εφαρμογές αυτής της κατηγορίας συμπεριλαμβάνουν την φωνητική αναζήτηση, τον προσωπικό ψηφιακό βοηθό (“What is Cortana,” 2017), το gaming, τα συστήματα δια δράσης για το σπίτι, τα συστήματα πληροφορίας και διασκέδασης για τα αυτοκίνητα. Παράλληλα υπάρχει και έντονο ενδιαφέρον αλλά και εφαρμογές για την χρήση της τεχνολογίας αυτής, ώστε να βοηθηθούν άτομα με ειδικές ανάγκες στην καθημερινότητα τους.

Οι εφαρμογές φωνητικής αναζήτησης (Wang, Yu, Ju, & Acero, 2008) επιτρέπουν στον χρήστη να ψάξει πληροφορίες για ενδιαφέροντα όπως, εστιατόρια, διευθύνσεις, προϊόντα κάνοντας χρήση του προφορικού λόγου μειώνοντας έτσι τον κόπο αλλά και τον χρόνο που απαιτείται για την ανεύρεση της πληροφορίας. Στην σημερινή εποχή, ο παραπάνω τρόπος αναζήτηση πληροφορίας είναι πολύ δημοφιλής σε όλες τις πλατφόρμες συστημάτων όπως το Android phone το iphone και το Windows Phone.



Εικόνα 18. Η εφαρμογή της Apple-siri
(<http://www.apple.com>)

Οι προσωπικοί ψηφιακοί βοηθοί παρότι υφίστανται εδώ και μια δεκαετία έγιναν τελευταία πολύ δημοφιλείς χάρη κυρίως του λανσαρίσματος του siri (Εικόνα 18) για iPhone από την Apple ("iOS - Siri - Apple," n.d.). Η εφαρμογή γνωρίζοντας τις πληροφορίες που έχουμε αποθηκεύσει στο κινητό μας, έχοντας πρόσβαση σε πληροφορίες στο διαδίκτυο και έχοντας καταγράψει κάποιες από τις συνήθειες ενέργειες μας, μπορεί

να μας εξυπηρετήσει πολύ καλύτερα. Έτσι για παράδειγμα με την χρήση του προφορικού λόγου μπορούμε να ψάξουμε για μουσική, να καλέσουμε κάποιον αριθμό να ρυθμίσουμε το πρόγραμμα επαφών μας.

Και στο gaming οι νέες παιχνιδομηχανές έχουν ενσωματώσει υπηρεσίες φωνής ώστε να μεγιστοποιήσουν την παρεχόμενη ευχαρίστηση προς τον χρήστη της συσκευής.

Τα συστήματα δια δράσης για το σπίτι, και τα συστήματα πληροφορίας και διασκέδασης για τα αυτοκίνητα (Seltzer, Ju, Tashev, Wang, & Yu, 2011) παρουσιάζουν πολλές ομοιότητες λειτουργικές, καθώς επιτρέπουν στον χρήστη με την χρήση προφορικού λόγου να παίζει μουσική, να αναζητήσει πληροφορίες να ελέγχει το σύστημα. Βέβαια το κάθε σύστημα έχει και τις δικές του σχεδιαστικές και κατασκευαστικές προκλήσεις. Σαν μια τέτοια μπορεί να θεωρηθεί ο σχεδιασμός εφαρμογών που θα εξυπηρετούν άτομα με ειδικές ανάγκες (Samah & Osman, 2015) καθώς ο σχεδιασμός θα πρέπει να προσαρμοστεί στις εκάστοτε απαιτήσεις του ατόμου.

Πέραν όμως τον δημοφιλών εφαρμογών, που στηρίζονται στην τεχνολογία της φωνής, για τον μέσο χρήστη, οι εξελίξεις σε αυτή την τεχνολογία βρίσκουν εφαρμογή και σε μια σειρά στρατιωτικών εφαρμογών. Ενδεικτικά μόνον μπορεί κανείς να αναφέρει την χρήση της παραπάνω τεχνολογίας στο Eurofighter Typhoon (Εικόνα 19) (“Eurofighter Typhoon | The world’s most advanced fighter aircraft,” 2017) αλλά και στο νεότερο και τεχνολογικά πολύ εξελιγμένο F 35.(Skaff, 2010) Και στις δυο περιπτώσεις οι φωνητικές εντολές έρχονται να βοηθήσουν τον χειριστή να έχει πιο καλό και γρήγορο έλεγχο πάνω στο πτητικό του μέσο. Οι συνθήκες φυσικά που βιώνουν οι πιλότοι μαχητικών είναι ακραίες ενώ το σώμα τους δέχεται δυνάμεις πολλών G. Το πως προσαρμόζεται ένα σύστημα φωνητικής αναγνώρισης και εντολών, όταν ο πιλότος είναι κάτω από την επίδραση πολλών G έχει διερευνηθεί (Englund, 2004) για το μαχητικό τέταρτης γενιάς JAS 39 Gripen (Εικόνα 20) με τα πρώτα αποτελέσματα να δίνουν πεδίο για περαιτέρω πειράματα και προσαρμογές του μηχανισμού φωνητικού ελέγχου. Αναλυτικά οι εφαρμογές της αυτόματης φωνητικής αναγνώρισης σε περιβάλλοντα στρατιωτικού χαρακτήρα θα παρουσιαστούν στο Κεφάλαιο



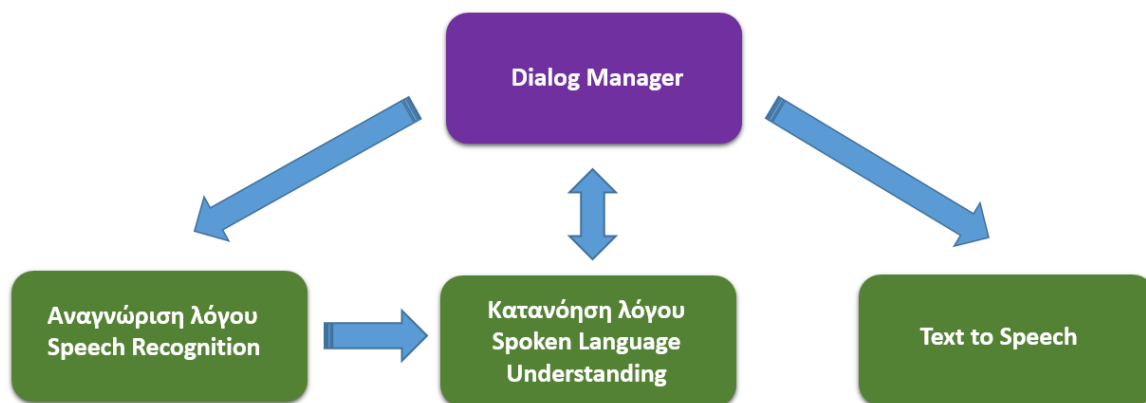
Εικόνα 19. Cockpit Eurofighter Typhoon
(<http://www.eurofighter.com/>)



Εικόνα 20. Το μαχητικό 4ης γενιάς JAS 39 Gripen
(<http://www.wikipedia.org>)

3.3 Αρχιτεκτονική και φιλοσοφία σχεδίασης

Όλα όσα προαναφέρθηκαν είναι ορισμένα μόνον παραδείγματα, από συστήματα προφορικού λόγου που βρίσκουν καθημερινή εφαρμογή. Όπως φαίνεται και στο Σχεδιάγραμμα 2 η φιλοσοφία σχεδιασμού των συστημάτων αυτών συνήθως εμπεριέχει από ένα η και παραπάνω κύρια μέρη.

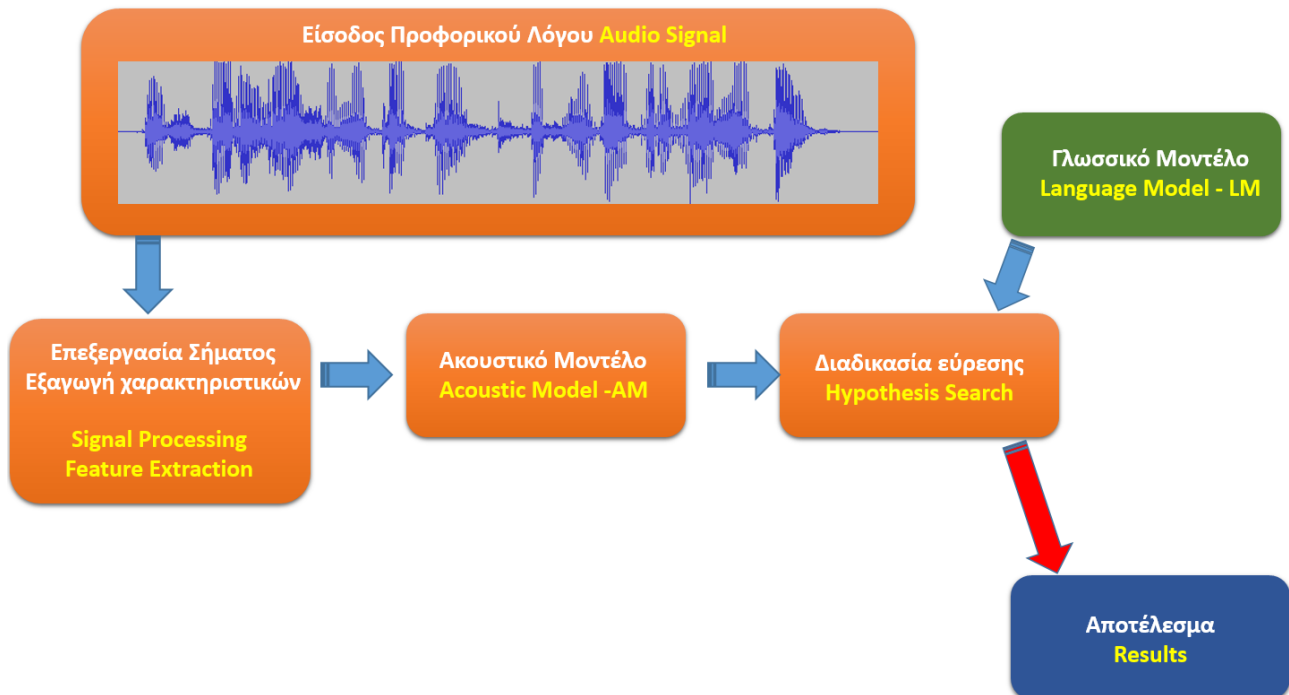


Σχεδιάγραμμα 2. Τα μέρη ενός τυπικού συστήματος προφορικού λόγου

Καταρχήν υπάρχει ένα μέρος αναγνώρισης του λόγου που μετατρέπει τον προφορικό λόγο σε κείμενο (speech recognition). Στην συνέχεια έχουμε ένα μέρος κατανόησης της ομιλούμενης γλώσσας (spoken language understanding) και ένα κείμενο προς λόγο μέρος (text to speech) που είναι υπεύθυνο για την μεταφορά της ομιλούμενης πληροφορίας. Τέλος υπάρχει και ένα μέρος που είναι υπεύθυνο για την «συνομιλία» (dialog manager) με τις εφαρμογές και τα άλλα τρία μέρη.

Η τυπική αρχιτεκτονική και φιλοσοφία σχεδίασης ενός συστήματος ASR φαίνεται στο σχεδιάγραμμα 3. Το σύστημα μας αποτελείται από τέσσερα μέρη. Από ένα μέρος επεξεργασίας σήματος και εξαγωγής χαρακτηριστικών, από το ακουστικό μοντέλο (Acoustic Model, AM), από το γλωσσικό μοντέλο (Language Model, LM) και από το μέρος που κάνει την διαδικασία εύρεσης (Hypothesis Search) αποδίδοντας στο τέλος το αποτέλεσμα σε μορφή κειμένου

Το πρώτο μέρος του συστήματος μας (επεξεργασία σήματος και εξαγωγή χαρακτηριστικών) λαμβάνει σαν είσοδο το ηχητικό σήμα. Σε πρώτη φάση «βελτιώνει» το σήμα αφαιρώντας τον θόρυβο και τυχόν διαταραχές ενώ στην συνέχεια μετατρέποντας το σήμα από τον χώρο του χρόνου στον χώρο της συχνότητας παράγει τα πιο χαρακτηριστικά διανύσματα που θα είναι κατάλληλα για το ακουστικό μοντέλο που ακολουθεί.



Σχεδιάγραμμα 3. Αρχιτεκτονική ενός συστήματος ASR

Το ακουστικό μοντέλο (Acoustic Model, AM) ενσωματώνει γνώσεις σχετικά με την ακουστική και τη φωνητική, παίρνει ως είσοδο τα χαρακτηριστικά που έχουν παραχθεί από το σύστημα επεξεργασίας και δημιουργεί μια βαθμολογία ακουστικού μοντέλου (AM score) για την ακολουθία χαρακτηριστικών μεταβλητού μήκους.

Το γλωσσικό μοντέλο (Language Model, LM) υπολογίζει την πιθανότητα μιας υποθετικής ακολουθίας λέξεων, δημιουργώντας την βαθμολογία του γλωσσικού μοντέλου (LM score). Αυτό είναι εφικτό με την εκμάθηση της συσχέτισης μεταξύ των λέξεων με βάση ένα τυπικό κείμενο που έχουμε χρησιμοποιήσει για τον σκοπό αυτό.

Τέλος το μέρος του συστήματος που κάνει την διαδικασία εύρεσης (Hypothesis search) συνδυάζει τις δύο βαθμολογίες που έχουν παραχθεί (AM score & LM score), δεδομένης της

ακολουθίας των χαρακτηριστικών διανυσμάτων και της ακολουθίας υποτιθέμενων λέξεων, και μας δίνει σαν αποτέλεσμα την ακολουθία λέξεων με την μεγαλύτερη βαθμολογία.

3.4 Κατηγορίες συστημάτων φωνητικής αναγνώρισης

Τα συστήματα φωνητικής αναγνώρισης μπορούν να κατηγοριοποιηθούν και να χαρακτηριστούν με βάση το τι εκφράσεις μπορούν να αναγνωρίσουν. Οι κατηγορίες που προκύπτουν είναι οι παρακάτω :

- **Συστήματα αναγνώρισης μεμονωμένων λέξεων.** Τα συστήματα που αναγνωρίζουν μεμονωμένες λέξεις χρειάζονται συνήθως μια περίοδο ησυχίας και στα δυο μέρη του παραθύρου δειγματοληψίας ενώ δέχονται μεμονωμένες λέξεις ή εκφράσεις κάθε φορά.
- **Συστήματα αναγνώρισης εκφράσεων.** Πρόκειται για συστήματα παρόμοια με τα παραπάνω τα οποία όμως επιτρέπουν και την ταυτόχρονη επεξεργασία εκφράσεων αρκεί αυτές να έχουν κάποιο ελάχιστο κενό ανάμεσα τους.
- **Συνεχής ομιλία.** Τα συστήματα αυτά επιτρέπουν στον ομιλητή να μιλά με συνεχή ροή λόγου και φυσικά. Είναι από τα πιο δύσκολα συστήματα στην δημιουργία καθώς εμπεριέχουν τεχνικές που καθορίζουν τα διάφορα όρια των εκφράσεων.
- **Αυθόρμητη ομιλία.** Μπορεί να θεωρηθεί ως ομιλία που είναι φυσική και χωρίς πρόβα. Ένα σύστημα φωνητικής αναγνώρισης αυθόρμητης ομιλίας θα πρέπει να είναι σε θέση να χειρίζεται μια ποικιλία από φυσικά χαρακτηριστικά της ανθρώπινης ομιλίας, λέξεις που χρησιμοποιούμε στην ομιλία μας όπως το «α» το «ε» κτλ.

Τα CMU Sphinx και Pocketsphinx μπορούν να ανταποκριθούν και στις τέσσερις παραπάνω περιπτώσεις αναγνώρισης του προφορικού λόγου.

3.5 Τεχνικές μοντελοποίησης

Οι διάφορες τεχνικές μοντελοποίησης σαν στόχο έχουν να δημιουργήσουν μοντέλα ομιλητών, χρησιμοποιώντας συγκεκριμένα με βάση των ομιλητή, χαρακτηριστικά. Η όλη τεχνική χωρίζεται στο στάδιο της αναγνώρισης ομιλητή και στο στάδιο της ταυτοποίησης του ομιλητή. Η τεχνική ταυτοποίησης ομιλητών προσδιορίζει αυτόματα ποιος μιλάει με βάση μεμονωμένες πληροφορίες που ενσωματώνονται στο σήμα ομιλίας. Ο κύριος στόχος της διαδικασίας ταυτοποίησης είναι να γίνει αυτή εφικτή συγκρίνοντας ένα άγνωστο ηχητικό σήμα με μια βάση ηχητικών σημάτων από γνωστούς ομιλητές. Τώρα ένα εκπαιδευμένο σύστημα, με μια σειρά διαφορετικών ομιλητών, είναι σε θέση να αναγνωρίσει τον ομιλητή. Παρακάτω, ολοκληρώνοντας και την ενότητα αυτή, θα αναφερθούμε στις διαφορετικές προσεγγίσεις που χρησιμοποιούνται, για την μοντελοποίηση των συστημάτων φωνητικής αναγνώρισης, όπως αυτές έχουν καταγραφεί στην εργασία των (Gaikwad, Gawali, & Yannawar, 2010).

3.5.1 Η ακουστικό-φωνητική προσέγγιση. Η μέθοδος αυτή έχει μελετηθεί για πάνω από 40 χρόνια και βασίζεται πάνω στην θεωρία των ακουστικών φωνητικών και των υποθέσεων (Hemdal & G. W. Hughes, 1967). Οι πρώτες προσεγγίσεις στην αναγνώριση ομιλίας βασίστηκαν στην εύρεση ήχων ομιλίας και στην συνέχεια στην παροχή κατάλληλων ετικετών σε αυτούς, κάτι που είναι και η βάση αυτής της προσέγγισης. Σύμφωνα με αυτή την προσέγγιση υποθέτουμε ότι στον προφορικό λόγο υπάρχουν πεπερασμένες, διακριτές φωνητικές μονάδες, τα φωνήματα, που χαρακτηρίζονται από ένα σύνολο ακουστικών ιδιοτήτων. Παρόλο που αυτές οι ιδιότητες είναι μεταβλητές θεωρούμε ότι οι κανόνες που διέπουν αυτή την μεταβλητότητα είναι σχετικά ξεκάθαροι και ότι μπορούν να μαθευτούν από μια υπολογιστική μηχανή.

3.5.2 Προσέγγιση με βάση την αναγνώριση μοτίβου (Rabiner & Juang, 1986). Η μέθοδος αυτή χωρίζεται σε δυο επιμέρους μέρη. Την εκπαίδευση του μοτίβου και την αναγνώριση του μοτίβου. Χρησιμοποιώντας ένα καλά διαμορφωμένο μαθηματικό πλαίσιο μπορεί να πετύχει μια αξιόπιστη σύγκριση προτύπων, από ένα σύνολο δειγμάτων κατάρτισης που έχουν προκύψει από τον αλγόριθμο εκπαίδευσης. Η αναπαράσταση ενός λεκτικού μοντέλου μπορεί να έχει την μορφή ενός λεκτικού προτύπου, η ενός στατιστικού μοντέλου (πχ. Κρυφό Μαρκοβιανό Μοντέλο-Hidden Markov Model-HMM) και μπορεί να εφαρμοστεί σε έναν ήχο μικρότερο από λέξη, σε λέξη ή σε φράση. Στην φάση σύγκρισης μοτίβου, της προσέγγισης αυτής, έχουμε μια σύγκριση ανάμεσα στον προφορικό λόγο που θέλουμε να αναγνωρίσουμε και σε κάθε δυνατόν υποθετικό μοτίβο που έχει προκύψει από την διαδικασία της εκπαίδευσης. Το αποτέλεσμα που παίρνουμε είναι το μοτίβο με την καλύτερη βαθμολογία κατά την φάση της σύγκρισης.

3.5.3 Προσέγγιση με βάση την αναγνώριση προτύπων (R.K.Moore, 1994). Σε αυτή την μεθοδολογία έχουμε την σύγκριση του άγνωστου λόγου με ένα σετ προ ηχογραφημένων λέξεων με στόχο να βρούμε την καλύτερη ταύτιση. Η βασική ιδέα της μεθοδολογίας είναι απλή. Μια συλλογή πρωτότυπων ομιλίας αποθηκεύεται ως πρότυπα αναφοράς που αντιπροσωπεύουν το λεξικό των υποψηφίων λέξεων. Στην συνέχεια η αναγνώριση του άγνωστου λόγου γίνεται με την ταύτιση του με κάθε ένα από τα πρότυπα αναφοράς συλλέγοντας την κατηγορία του πιο ταυτιζόμενου προτύπου.

3.5.4 Δυναμική χρονική στρέβλωση (Dynamic time warping-DTW) (Senin, 2008). Η δυναμική χρονική στρέβλωση είναι ένας αλγόριθμος που μετράει ομοιότητες ανάμεσα σε δύο αλληλουχιών που μπορεί να διαφέρουν στον χρόνο ή την ταχύτητα. Έχει εφαρμοστεί με επιτυχία σε δεδομένα γραφικών, video ή ήχου και γενικά τα δεδομένα που μπορούν να έχουν μια γραμμική αναπαράσταση μπορούν να δεχθούν επεξεργασία με χρήση του συγκεκριμένου αλγορίθμου. Ο αλγόριθμος επιτρέπει σε ένα υπολογιστικό σύστημα να βρει την ιδεατή αντιστοίχιση ανάμεσα σε δυο διαφορετικές αλληλουχίες ,πάντα στο πλαίσιο συγκεκριμένων περιορισμών. Ένα παράδειγμα των περιορισμών που επιβάλλονται στην αντιστοίχιση των ακολουθιών είναι η μονοτονικότητα της χαρτογράφησης στην χρονική διάσταση. Η συνέχεια είναι λιγότερο σημαντική στον DTW από ότι σε άλλους αλγόριθμους αντιστοίχισης προτύπων. Τέλος ο DTW είναι ένας αλγόριθμος ιδιαίτερα κατάλληλος για την αντιστοίχιση αλληλουχιών με ελλείπουσες πληροφορίες, υπό την προϋπόθεση ότι υπάρχουν αρκετά μεγάλα τμήματα για την επίτευξη αντιστοίχισης.

3.5.5 Προσέγγιση με βάση την υπαρκτή γνώση (Gaikwad et al., 2010). Εδώ έχουμε την δημιουργία μιας βάσης γνώσης για μια γλώσσα η οποία στην συνέχεια μεταφέρεται κωδικοποιημένη σε μια μηχανή. Αυτό θεωρητικά μας δίνει την δυνατότητα για μια πολύ καλή μοντελοποίηση των διαφόρων εκφάνσεων μια λεκτικής διαλέκτου. Στην πράξη όμως το να μπορέσεις να αποκτήσεις αυτή την γνώση που θα καλύπτει όλες τις περιπτώσεις μιας γλώσσας και στην συνέχεια να την χρησιμοποιήσεις με επιτυχία δεν είναι και τόσο εύκολα εφικτό. Η συγκεκριμένη προσέγγιση κρίνεται, εκ των αποτελεσμάτων, μη πρακτική και επιδιώκεται η αυτόματη διαδικασία εκπαίδευσης και μάθησης.

3.5.6 Προσέγγιση με βάση στατιστικά μοντέλα. Κατά την προσέγγιση αυτή ο προφορικός λόγος μοντελοποιείται στατιστικά με την χρήση αυτόματων εκπαιδευόμενων στατιστικών διαδικασιών συνήθως των Κρυφών Μαρκοβιανών Μοντέλων (Rabiner & Juang, 1986) (Hidden Markov Models-HMM). Το κύριο μειονέκτημα των στατιστικών μοντέλων είναι ότι πρέπει να λαμβάνουν προκαταρκτικά υποδείγματα μοντελοποίησης, τα οποία είναι ανακριβή, μειώνοντας έτσι την απόδοση του συστήματος. Στα συστήματα αναγνώρισης φωνής τα HMM έχουν εφαρμοστεί με μεγάλη επιτυχία. Άλλωστε και η πλατφόρμα CMU Sphinx,

που αποτελεί και θέμα εργασίας αυτής της Μεταπτυχιακής διατριβής, στηρίζεται στην μοντελοποίηση, του προφορικού λόγου, με την χρήση Κρυφών Μαριοβιανών Μοντέλων.

3.5.7 Εφαρμογές με βάση την εκπαίδευση (Gaikwad et al., 2010). Για να ξεπεραστούν τα μειονεκτήματα των εφαρμογών που κάνουν χρήση των HMM, μπορούν να εισαχθούν μέθοδοι εκμάθησης όπως τα νευρωνικά δίκτυα και ο προγραμματισμός γενετικού αλγορίθμου. Σε αυτά τα μοντέλα μάθησης δεν είναι απαραίτητο να δοθούν σαφείς κανόνες ή άλλες γνώσεις. Αυτό μπορεί να γίνει αυτόματα μέσω προσομοιώσεων ή μιας εξελικτικής διαδικασίας.

3.5.8 Προσέγγιση με βάση την Τεχνητή Νοημοσύνη (Gaikwad et al., 2010). Σε αυτές τις εφαρμογές έχουμε την προσπάθεια να μηχανοποιήσουμε την διαδικασία αναγνώρισης της φωνής. Αυτό γίνεται με βάση το πώς ένα άτομο εφαρμόζει την ευφυΐα του στην οπτικοποίηση ανάλυση και λήψη απόφασης πάνω σε μετρημένα ακουστικά χαρακτηριστικά. Πρόκειται για μια υβριδική μεθοδολογία ανάμεσα στην ακουστικό-φωνητική προσέγγιση και στην προσέγγιση αναγνώρισης προτύπων. Η βαθιά γνώση που έχει πλέον επιτευχθεί σχετικά με την διαδικασία της ανθρώπινης ομιλίας έχει βοηθήσει στον σχεδιασμό εφαρμογών που κάνουν χρήση Τεχνητής Νοημοσύνης. Ωστόσο, η προσέγγιση αυτή είχε περιορισμένη επιτυχία, κυρίως λόγω της δυσκολίας ποσοτικοποίησης των γνώσεων που υπάρχουν. Ένα άλλο δύσκολο πρόβλημα είναι η ενσωμάτωση πολλών επιπέδων ανθρώπινης φωνητικής γνώσης, φωνοτακτικής, λεξικής πρόσβασης, σύνταξης, και σημασιολογίας. Δυστυχώς, ο συνδυασμός ανεξάρτητων και ασύγχρονων πηγών λεκτικής γνώσης εξακολουθεί να αποτελεί ένα πολύ δύσκολο αν όχι άλυτο πρόβλημα.

3.5.9 Στοχαστική προσέγγιση (Gaikwad et al., 2010). Σε αυτή την μεθοδολογία χρησιμοποιούμε μοντέλα πιθανοτήτων για να αντιμετωπίσουμε αβέβαιες ή ελλιπείς πληροφορίες. Καθώς στις εφαρμογές αναγνώρισης φωνής η αβεβαιότητα και η ατέλεια είναι υπαρκτές (και μάλιστα από πολλές πηγές) τα στοχαστικά μοντέλα είναι μια κατάλληλη προσέγγιση. Η πιο δημοφιλής στοχαστική προσέγγιση είναι τα Κρυφά Μαριοβιανά Μοντέλα (HMM), που χαρακτηρίζονται από ένα από μια πεπερασμένη κατάσταση και ένα σύνολο κατανομών εξόδου.

3.5.10 Προσέγγιση με βάση τα Βαθιά Νευρωνικά Δίκτυα (DNN Deep neural network) (Hinton et al., 2012). Τα τελευταία χρόνια, τα διακριτά ιεραρχικά μοντέλα όπως τα βαθιά νευρωνικά δίκτυα έγιναν εφικτά και με τον την εφαρμογή τους είχαμε σημαντική μείωση των ποσοστών σφάλματος χάρη στην συνεχιζόμενη εξέλιξη στην υπολογιστική ισχύ, στην διαθεσιμότητα μεγάλων σετ για εκπαίδευση αλλά και στην καλύτερη κατανόηση των μοντέλων αυτών (Yu, 2014). Ουσιαστικά πρόκειται για τεχνητά νευρωνικά δίκτυα τα οποία

έχουν παραπάνω από ένα «στρώματα» από κρυμμένες μονάδες μεταξύ των εισόδων και των εξόδων τους. Κάθε κρυμμένη μονάδα συνήθως χρησιμοποιεί την λογιστική λειτουργία για να ταιριάζει την συνολική είσοδο από το παρακάτω επίπεδο με την βαθμωτή κατάσταση που στέλνει στο από πάνω στρώμα. Τα βαθιά νευρωνικά δίκτυα που αποτελούνται από πολλά κρυμμένα στρώματα και πολλές μονάδες ανά στρώμα είναι πολύ ευέλικτα και με έναν μεγάλο αριθμό παραμέτρων (Hinton et al., 2012). Αυτό τα κάνει ικανά για την μοντελοποίηση πολύπλοκων μη-γραμμικών σχέσεων μεταξύ εισόδων και εξόδων. Αυτή τους η ικανότητα είναι πολύ σημαντική για υψηλής ποιότητας και ακρίβειας ακουστική μοντελοποίηση. Για να αποφύγουμε το πρόβλημα του υπερ-ταιριάσματος χρησιμοποιούμε ποινές με αντίστοιχο βάρος αλλά αυτό μας αφαιρεί σημαντική δυναμική κατά την μοντελοποίηση. Με την χρήση μεγάλων σετ εκπαίδευσης μπορούμε να αντιμετωπίσουμε το πρόβλημα αυτό αλλά αυτό θα μας «κοστίσει» σε υπολογιστική ισχύ.

ΚΕΦΑΛΑΙΟ 2

Οι πλατφόρμες αυτόματης φωνητικής αναγνώρισης CMU Sphinx και Pocketsphinx

§1. Γενικά

Η πλατφόρμα αυτόματης φωνητικής αναγνώρισης CMU Sphinx έχει αναπτυχθεί από κοινού από το Πανεπιστήμιο Carnegie Mellon τα εργαστήρια της Sun Microsystems και τα ερευνητικά εργαστήρια της Mitsubishi. (Lamere et al., 2003). Έχει αναπτυχθεί με την χρήση της γλώσσας προγραμματισμού Java, και αποτελεί μια πολύ ευέλικτη πλατφόρμα που υποστηρίζει όλους τους τύπους ακουστικών μοντέλων που είναι βασισμένοι σε Κρυφά Μαρκοβιανά Μοντέλα (HMM). Πρόκειται για μια προσπάθεια ανοιχτού κώδικα, ο κώδικας της οποίας βρίσκεται διαθέσιμος για όλους στον ιστότοπο της SourceForge.

Η ανάπτυξη της πλατφόρμας CMU Sphinx έγινε στη Java η οποία προσφέρει συγκριτικά πλεονεκτήματα, κυρίως στον τομέα συντήρησης του κώδικα. Επίσης η Java προσφέρει την δυνατότητα στα προγράμματα που έχουν αναπτυχθεί με την χρήση της, να είναι εκτελέσιμα σε διαφορετικά λειτουργικά συστήματα. Ταυτόχρονα είναι πιο εύκολη η εκμετάλλευση των δυνατοτήτων που δίνουν τα υπολογιστικά μηχανήματα, με πολλούς επεξεργαστές, επιτρέποντας μας έτσι ιδίως στην διαδικασία της εκπαίδευσης του μοντέλου μας να έχουμε μεγάλα σετ εκπαιδευτικών δεδομένων.

Το CMU Sphinx έχει σχεδιαστεί με τέτοιο τρόπο ώστε να αποτελείται από διαφορετικά μέρη στα οποία μπορούμε να κάνουμε αλλαγές χωρίς να επηρεάζεται η συνολική αρχιτεκτονική του συστήματος μας. Έτσι για παράδειγμα μπορούμε να επιλέξουμε διαφορετικούς τύπους γλωσσικών μοντέλων μόνον με την απλή παραμετροποίηση σε ένα μέρος του συστήματος μας.

Το Pocketsphinx φιλοδοξούσε να αποτελέσει μια απλή και μη απαιτητική έκδοχή του CMU Sphinx για φορητές συσκευές (Huggins-Daines et al., 2006) καθώς είχε ήδη ξεκινήσει η ραγδαία ανάπτυξη των φορητών συσκευών και εφαρμογών. Πλην όμως έγινε φανερό από την

αρχή ότι για να μπορέσει το Pocketsphinx να έχει μια λογική απόδοση σαν εφαρμογή χρειαζόταν μια νέα αρχιτεκτονική σχεδιασμού. Σε αντίθεση με το CMU Sphinx το Pocketsphinx αναπτύχθηκε σε γλώσσα προγραμματισμού ANSI C ενώ οι στόχοι που ορίστηκαν κατά τον σχεδιασμό της αρχιτεκτονικής του από τον (Daines, 2011) ήταν :

- Φορητότητα της εφαρμογής
- Απλή υλοποίηση
- Αποτύπωμα μικρού κώδικα και δεδομένων
- Ασφάλεια
- Απόδοση μνήμης

Το Pocketsphinx μπορεί να χρησιμοποιηθεί λόγω των χαρακτηριστικών του σε μια σειρά συσκευών, και προτιμήθηκε κατά τις εργασίες της Μεταπτυχιακής διατριβής, καθώς αυτές έγιναν με ένα υπολογιστή μέτριας υπολογιστικής ισχύος.

Στην συνέχεια του κεφαλαίου θα παρουσιαστούν τα βασικά χαρακτηριστικά και η δομή του προφορικού λόγου καθώς και οι βασικές αρχές λειτουργίας του Sphinx. Αναλυτική παρουσίαση των αρχιτεκτονικών σχεδιασμού τόσο του Sphinx όσο και του Pocketsphinx θα γίνει στο τρίτο μέρος του κεφαλαίου πριν περάσουμε στην περιγραφή των εργασιών δημιουργίας του Ελληνικού μοντέλου για την πλατφόρμα CMU Sphinx.

§2. Δομή και βασικά χαρακτηριστικά του προφορικού λόγου

2.1 Γενικά

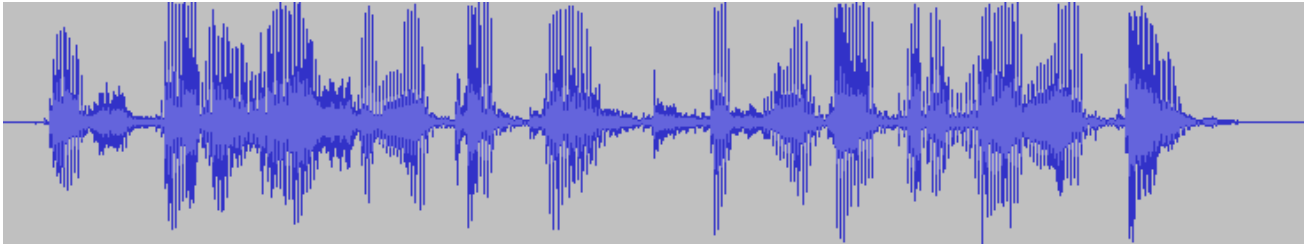
Χωρίς την γλώσσα, τον προφορικό λόγο δηλαδή, δεν μπορεί να νοηθεί ζωή και οι σημαντικότερες εκφάνσεις της. Είτε θεωρήσουμε τον προφορικό λόγο βιολογικό φαινόμενο είτε μια πνευματική ή κοινωνική εκδήλωση. Για αυτό και η πρόοδος του ανθρώπου περνάει μέσα από την γλωσσική του καλλιέργεια. Ένας κώδικας σημείων ορισμένης γλωσσικής μορφής με τα οποία επιτυγχάνεται η επικοινωνία μεταξύ των μελών μιας γλωσσικής κοινότητας. Όπως έχει αναλυθεί στο Κεφάλαιο 1 η κύρια λειτουργία του προφορικού λόγου είναι η επικοινωνία. Με τον συμβολικό χαρακτήρα που έχουν οι λέξεις, τα αντικείμενα γίνονται έννοιες. Οι έννοιες αποθηκεύονται στην μνήμη μας, ανακαλούνται από αυτήν συντάσσονται και εννοούνται με συμβατικό τρόπο που επιβάλλει η ανθρώπινη λογική μας και η κυριολεξία του υποσυνείδητου μας.

Η σημασία που έχει η επικοινωνία μέσω του προφορικού λόγου προσδιορίζει σαφώς το πόσο πολύτιμη για την ζωή είναι η λειτουργία αυτή. Χάρη στον προφορικό λόγο οργανώθηκαν κοινωνίες, μεταδόθηκαν εμπειρίες, ανταλλάχθηκαν απόψεις και γνώσεις, εκφράστηκαν συναισθήματα, περιεγράφηκαν γεγονότα δημιουργήθηκε πολιτισμός.

2.2 Η δομή και τα χαρακτηριστικά του προφορικού λόγου

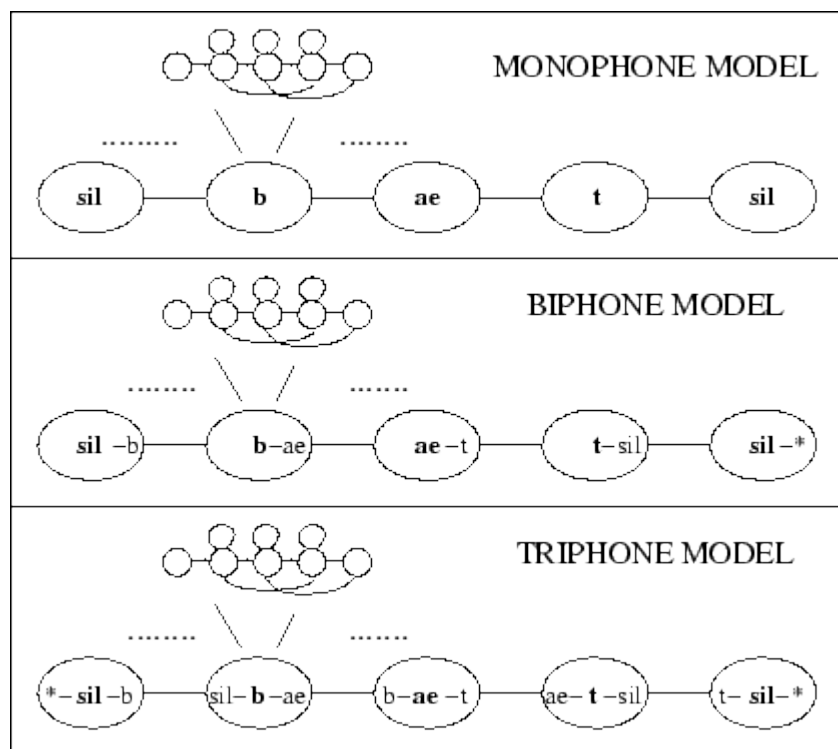
Ο προφορικός λόγος είναι ένα σύνθετο φυσικό φαινόμενο το οποίο οι πιο πολλοί από μας δύσκολα αντιλαμβανόμαστε πως παράγεται. Η άποψη που κυριαρχεί είναι ότι ο προφορικός λόγος αποτελείται από λέξεις και κάθε λέξη είναι ένα σύνολο φωνημάτων. Η πραγματικότητα όμως είναι διαφορετική (Young, 2008). Ο προφορικός λόγος είναι μια δυναμική διεργασία χωρίς πολύ ξεκάθαρα χωρισμένα επιμέρους μέρη. Μια εξέταση της κυματικής μορφής μιας ηχογράφησης (Εικόνα 21) όπως αυτή έχει αποτυπωθεί σε ένα πρόγραμμα ηχογράφησης και επεξεργασίας ήχου (<http://www.audacityteam.org/>) μας βοηθά να κατανοήσουμε καλύτερα την φυσική μορφή του προφορικού λόγου. Η δομή του προφορικού λόγου μπορεί να κατανοηθεί όπως παρακάτω. Ο προφορικός λόγος είναι μια συνεχής εκπομπή ηχητικών σημάτων όπου αρκετά σταθερές καταστάσεις αναμιγνύονται με δυναμικά αλλαγμένες καταστάσεις. Σε αυτή την αλληλουχία των καταστάσεων μπορούμε να αναγνωρίσουμε ομάδες ήχων με τα ίδια ή παρόμοια φωνητικά χαρακτηριστικά που μπορούμε να τα ονομάσουμε φωνήματα (phonemes).

Βέβαια τα φωνητικά χαρακτηριστικά αυτά των φωνημάτων εξαρτώνται από παράγοντες όπως ο ομιλητής, το είδος του προφορικού λόγου (αργό η γρήγορο) το ύψος του προφορικού λόγου (απαθές, συναισθηματικά φορτισμένο κτλ.). Επίσης το συντακτικό των φωνημάτων τα κάνει να ακούγονται εντελώς διαφορετικά από την κανονική τους ηχητική παράσταση τους.



Εικόνα 21. Κυματική μορφή προφορικού λόγου

Σαν διπλά φωνήματα ορίζουμε τις περιοχές μεταξύ δύο διαδοχικών φωνημάτων ενώ



μερικές φορές μπορούμε να αναφερθούμε σε υποφωνηματικές μονάδες που αποτελούν υπο καταστάσεις ενός φωνήματος. Συχνά βέβαια μπορούμε να βρούμε τρεις η παραπάνω περιοχές διαφορετικής φύσεως ενός φωνήματος. Στην αυτόματη φωνητική αναγνώριση συχνά επιλέγουμε να χρησιμοποιήσουμε τις τρεις καταστάσεις ενός φωνήματος (Εικόνα 22). Την πρώτη που εξαρτάται από το φώνημα που προηγείται, την μεσαία που είναι η σταθερή κατάσταση και την επόμενη που εξαρτάται από το φώνημα που ακολουθεί. Αρκετές φορές ορίζουμε ένα διαφορετικό πλαίσιο για τα φωνήματα το οποίο εξαρτάται

Εικόνα 22. Απλό διπλό και τριπλό φώνημα της Αγγλικής λέξης bat
(<http://www.igi.tugraz.at>)

από την γειτνίαση μεταξύ τους. Κωδικοποιώντας αλληλουχίες που αποτελούνται από τρία

φωνήματα (τα τριπλά φωνήματα) επιλέγουμε κατά την φάση της φωνητικής αναγνώρισης να εντοπίζουμε μέρη των τριπλών φωνημάτων παρά να τα εντοπίζουμε ολόκληρα.

Έτσι δημιουργούμε «εντοπιστές» για τις αρχές των τριπλών φωνημάτων και καταλήγουμε να έχουμε μια εργαλειοθήκη από 4000 περίπου «εντοπιστές» χαρακτηριστικών ήχων. Αυτοί οι «εντοπιστές» ήχων ονομάζονται *senones*. Τα φωνήματα δημιουργούν υπό λεκτικές μονάδες τις συλλαβές. Οι δε συλλαβές δημιουργούν τις λέξεις. Οι λέξεις είναι πολύ σημαντικές για τις εργασίες αυτόματης φωνητικής αναγνώρισης καθώς περιορίζουν σημαντικά τους λογικούς συνδυασμούς φωνημάτων. Επίσης ένας παράγοντας που βοηθά σημαντικά στις προσπάθειες κατά την διαδικασία της αυτόματης φωνητικής αναγνώρισης, είναι το γεγονός ότι ακόμα και οι πιο μορφωμένοι ομιλητές δεν χρησιμοποιούν παραπάνω από 20000 λέξεις κατά την ομιλία τους. Αυτό περιορίζει σημαντικά το υπολογιστικό έργο που έχουμε να κάνουμε και που έτσι και αλλιώς είναι αρκετά φορτωμένο.

Οι λέξεις και μια σειρά μη λεκτικών ήχων, όπως οι κομπασμοί, δημιουργούν μαζί τις εκφράσεις που αποτελούν ξεχωριστά μέρη μεταξύ των παύσεων. Θα πρέπει να αναφέρουμε εδώ ότι οι εκφράσεις δεν ταιριάζουν απαραίτητα με προτάσεις.

2.3 Η μεθοδολογία αναγνώρισης του CMU Sphinx

Στο σημείο αυτό θα δοθεί το γενικό πλαίσιο των διαδικασιών που χρησιμοποιούνται από το CMU Sphinx, κατά την διαδικασία της αυτόματης φωνητικής αναγνώρισης, πριν περάσουμε στην συνέχεια στην παρουσίαση των κυρίως μερών και αρχιτεκτονικών τόσο του CMU Sphinx όσο και του Pocketsphinx.

Ο συνήθης τρόπος για να αναγνωρίσει κανείς φωνή (προφορικό λόγο) είναι ο ακόλουθος. Παίρνουμε την κυματική μορφή του λόγου, την χωρίζουμε σε εκφράσεις σύμφωνα με τα σημεία των παύσεων, και μετά προσπαθούμε να αναγνωρίσουμε τι λέγεται σε κάθε έκφραση. Για να το καταφέρουμε αυτό θα πρέπει να πάρουμε όλους τους δυνατούς συνδυασμούς λέξεων και να προσπαθήσουμε να τους ταιριάξουμε με το ηχητικό μας δείγμα. Σε τι θα πρέπει τώρα να δώσουμε σημασία κατά την διαδικασία αυτή.

Καταρχήν η όλη διαδικασία έχει μέσα της την έννοια των χαρακτηριστικών. Καθώς ο αριθμός των παραμέτρων της διαδικασίας είναι μεγάλος, προσπαθούμε να βελτιστοποιήσουμε την διαδικασία. Οι αριθμοί που συνήθως υπολογίζονται από τον λόγο προκύπτουν από τον χωρισμό του σε καρέ. Μετά από κάθε καρέ που συνήθως διαρκεί 10 millisecond υπολογίζουμε 39 αριθμούς που αντιπροσωπεύουν τον λόγο και ονομάζεται διάνυσμα χαρακτηριστικών. Ο τρόπος υπολογισμού αυτών των 39 αριθμών είναι πεδίο διαρκούς έρευνας και πειραματισμού.

Στην συνέχεια έχουμε την έννοια του μοντέλου. Ουσιαστικά πρόκειται για περιγραφή ενός μαθηματικού αντικειμένου, που συγκεντρώνει κοινά χαρακτηριστικά της ομιλούμενης λέξης. Στην πράξη το ακουστικό μοντέλο ενός «εντοπιστή» (senone) για παράδειγμα είναι η γκαουσιανή μίξη των τριών καταστάσεων του. Τίθενται βεβαίως τα ερωτήματα πόσο καλό είναι ένα μοντέλο, πόσο καλά μπορεί να προσαρμοστεί σε νέα δεδομένα σε αλλαγές κτλ. Το ακουστικό μοντέλο που χρησιμοποιείται στην οικογένεια προγραμμάτων Sphinx βασίζεται πάνω στα Κρυφά Μαρκοβιανά Μοντέλα (HMM) τα οποία έχουν αποδειχθεί πολύ αποτελεσματικά για την αποκωδικοποίηση κατά την φάση της αυτόματης φωνητικής αναγνώρισης (Karyotis, 2016). Εδώ ο προφορικός λόγος περιγράφεται σαν μια αλληλουχία καταστάσεων που αλλάζουν ή μια προς την άλλη ακολουθώντας μια πιθανοτική λογική.

Τέλος είναι η διαδικασία της ταυτοποίησης του λόγου. Προφανώς το να συγκρίνεις όλα τα διανύσματα χαρακτηριστικών με τα ακουστικά μοντέλα είναι τρομερά κοστοβόρο σε χρόνο και υπολογιστική δύναμη. Έτσι με μια σειρά διαδικασιών προσπαθούμε να βελτιστοποιήσουμε το ταίριασμα αυτό και σε κάθε σημείο κρατάμε πάντα τις πιο επιτυχημένες επιλογές που έχουν προκύψει.

Κατά τις εργασίες που διενεργούνται συνεπώς με την πλατφόρμα εργαλείων CMU Sphinx, και ακολουθώντας την γενική φιλοσοφία των προγραμμάτων αυτόματης αναγνώρισης που έχει ήδη παρουσιαστεί, τρία επιμέρους μοντέλα χρησιμοποιούνται ώστε να πετύχουμε την αναγνώριση.

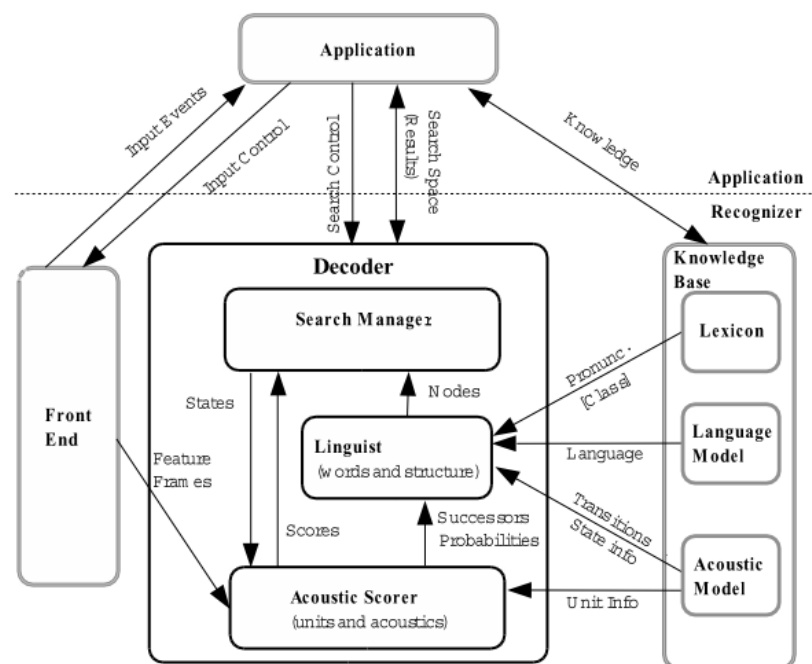
- Το Ακουστικό μοντέλο (Acoustic model) που είναι μια σειρά αρχείων που περιέχουν στατιστικές πληροφορίες σχετικά με βασικές μονάδες του λόγου (senones).
- Το Γλωσσικό μοντέλο (Language model) που είναι ένα αρχείο που περιγράφει την γλώσσα που θέλουμε να αναγνωρίσουμε. Μας βοηθά στο να περιορίσουμε τις αναζητήσεις λέξεων που κάνουμε, καθώς ορίζει ποια λέξη πιθανόν να ακολουθεί μιας ήδη αναγνωρισμένης λέξης.
- Το φωνητικό λεξικό (Phonetic dictionary) που περιέχει την αντιστοίχιση των λέξεων μας σε μια μορφή, κατανοητή από την πλατφόρμα η οποία περιέχει φωνήματα.

Τα προαναφερθέντα μοντέλα, τα οποία και θα παρουσιασθούν αναλυτικά παρακάτω, δεν υπήρχαν για την Ελληνική γλώσσα, και η δημιουργία τους αποτέλεσε και τον κύριο στόχο της Μεταπτυχιακής διατριβής.

§3. Αναλυτική περιγραφή αρχών λειτουργίας και αρχιτεκτονικής CMU Sphinx και rocketsphinx.

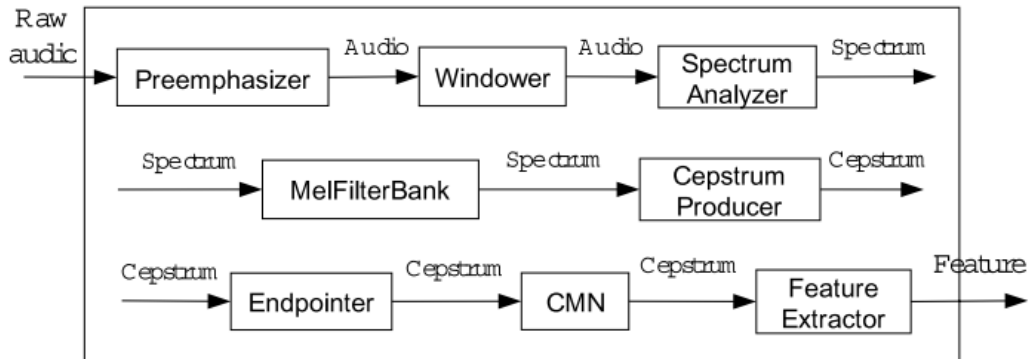
3.1 Αρχιτεκτονική λειτουργίας του CMU Sphinx

Η Αρχιτεκτονική λειτουργίας του CMU Sphinx (Lamere et al., 2003) φαίνεται αναλυτικά στο Σχεδιάγραμμα 4. Τα τρία κύρια μπλοκ του συστήματος είναι το front end το μπλοκ του αποκωδικοποιητή (decoder) και το μπλοκ της βάσης γνώσης (knowledge base). Ο αποκωδικοποιητής εμπεριέχει με την σειρά του τον γλωσσολόγο, τον διαχειριστή αναζήτησης και τον ακουστικό βαθμολογητή ενώ η βάση γνώσης εμπεριέχει το Ακουστικό και Γλωσσικό μοντέλο καθώς και το φωνητικό λεξικό. Στο σχεδιάγραμμα επίσης παρουσιάζονται όλες οι μεταξύ των τριών μπλοκ επικοινωνίες αλλά και μεταξύ του όλου συστήματος και εφαρμογών. Στην συνέχεια θα γίνει μια πιο αναλυτική περιγραφή του κάθε συστήματος ώστε να είναι εφικτή η καλύτερη κατανόησή της λειτουργίας του CMU Sphinx.



Σχεδιάγραμμα 4. Αρχιτεκτονική λειτουργίας CMU Sphinx
(Lamere et al., 2003)

3.1.1 Το front end μπλοκ



Σχεδιάγραμμα 5. Αρχιτεκτονική λειτουργίας του front end μπλοκ
(Lamere et al., 2003)

Στο Σχεδιάγραμμα 5 παρουσιάζεται η αρχιτεκτονική του front end μπλοκ του CMU Sphinx. Το συνολικό μπλοκ αποτελείται από αρκετά επικοινωνούντα αναμεταξύ τους μπλοκ, τα οποία έχουν το κάθε ένα μια είσοδο και μια έξοδο. Όταν ένα μπλοκ είναι έτοιμο να δεχτεί επιπλέον δεδομένα, διαβάζει από το προηγούμενο μπλοκ και επεξεργάζεται την πληροφορία ώστε να διαπιστώσει αν πρόκειται για δεδομένα φωνής ή για σήμα ελέγχου. Το σήμα ελέγχου μπορεί να ορίσει την έναρξη ή λήξη της ομιλίας, κάτι πολύ σημαντικό για τον αποκωδικοποιητή, ή δεδομένα τα οποία έχουν απορριφθεί ή κάποιο άλλο πρόβλημα. Αν τα εισερχόμενα δεδομένα είναι ομιλία τότε αυτά επεξεργάζονται και αποθηκεύονται προσωρινά μέχρι το ερχόμενο μπλοκ να τα ζητήσει για την δική του επεξεργασία.

Ένα από τα σχεδιαστικά χαρακτηριστικά του προγράμματος είναι το γεγονός ότι όλες οι έξοδοι των μπλοκ μπορούν να χρησιμοποιηθούν ενώ ταυτόχρονα δεν είναι απαραίτητο η είσοδος του συστήματος να είναι κατά ανάγκη αποκλειστικά η είσοδος του πρώτου μπλοκ. Επίσης υπάρχει δυνατότητα οποιοδήποτε από τα μπλοκ να αντικατασταθεί ενώ είναι δυνατόν η παρεμβολή ενδιάμεσων μπλοκ ανάμεσα σε δύο, ώστε για παράδειγμα να έχουμε μείωση του περιβάλλοντος θορύβου κτλ. Τα χαρακτηριστικά που υπολογίζονται χρησιμοποιώντας ανεξάρτητες πηγές πληροφοριών, όπως οπτικά χαρακτηριστικά, μπορούν να τροφοδοτούνται απευθείας στον αποκωδικοποιητή, είτε παράλληλα με τα χαρακτηριστικά από το σήμα ομιλίας, είτε παρακάμπτοντας όλα αυτά μαζί. Το όλο σύστημα είναι ικανό να «τρέχει» συνεχώς καθώς το τροφοδοτούμε με συνεχόμενα δεδομένα ομιλίας ενώ έχει την δυνατότητα να

λειτουργήσει και σε τρεις άλλες μορφές λειτουργίας που μας δίνουν περισσότερη ευελιξία κατά την χρήση του.

3.1.2 Ο αποκωδικοποιητής (decoder)

Ο αποκωδικοποιητής εμπεριέχει τρεις επιμέρους μονάδες τον γλωσσολόγο, τον διαχειριστή αναζήτησης και τον ακουστικό βαθμολογητή που περιγράφονται παρακάτω.

3.1.2.1 Διαχειριστής αναζήτησης. Η κύρια λειτουργία του διαχειριστή αναζήτησης είναι να κατασκευάσει και να αναζητήσει ένα δέντρο πιθανοτήτων για την καλύτερη υπόθεση. Η κατασκευή του δέντρου αναζήτησης γίνεται με βάση πληροφορίες που προέρχονται από τον γλωσσολόγο. Επιπλέον, ο διαχειριστής αναζήτησης επικοινωνεί με τον ακουστικό βαθμολογητή για να αποκτήσει τις βαθμολογίες για τα εισερχόμενα ηχητικά δεδομένα. Οι εργασίες του πραγματοποιούνται σύμφωνα με τους αλγόριθμους Viterbi (Viterbi, 1967) και Bushderby (Singh, Warmuth, Raj, & Lamere, 2003).

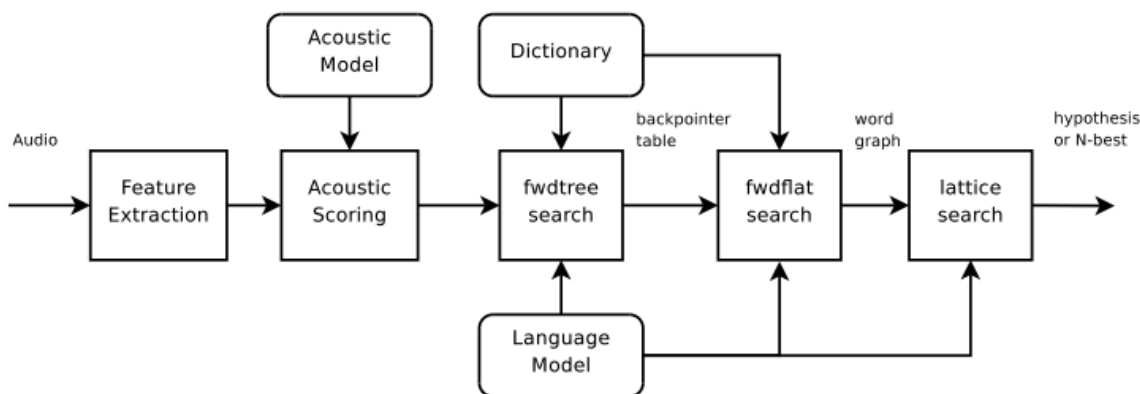
3.1.2.2 Γλωσσολόγος. Ο γλωσσολόγος δημιουργεί το γράφημα αναζήτησης που ο αποκωδικοποιητής χρησιμοποιεί κατά την διάρκεια της αναζήτησης των λέξεων, ενώ ταυτόχρονα κρύβει τις πολυπλοκότητες που σχετίζονται με τη δημιουργία αυτού του γραφήματος (Walker et al., 2004). Μια τυπική εφαρμογή γλωσσολόγου κατασκευάζει το γράφημα αναζήτησης χρησιμοποιώντας την δομή της γλώσσας όπως αυτή αντιπροσωπεύεται από ένα δεδομένο Γλωσσικό μοντέλο και την δομή του Ακουστικού μοντέλου. Ο γλωσσολόγος μπορεί επίσης να χρησιμοποιήσει ένα λεξικό (συνήθως ένα λεξικό προφοράς) για να αντιστοιχίσει λέξεις από το Γλωσσικό μοντέλο σε ακολουθίες στοιχείων του Ακουστικού μοντέλου. Κατά την δημιουργία του γραφήματος αναζήτησης ο γλωσσολόγος μπορεί να ενσωματώνει μονάδες υπό-λέξεων με πλαίσια αυθαίρετου μήκους εφόσον αυτά παρέχονται. Ο σχεδιασμός μας επιτρέπει να παρέχουμε διαφορετικές διαμορφώσεις για διαφορετικά συστήματα και απαιτήσεις αναγνώρισης επιτρέποντας μας έτσι να κάνουμε χρήση της αντίστοιχης διαμόρφωσης ανάλογα με την δουλειά που θέλουμε να κάνουμε. Έτσι για παράδειγμα μια απλή εφαρμογή αναγνώρισης αριθμών μπορεί να κάνει χρήση ενός απλού γλωσσολόγου που κρατά όλη την διαδικασία ανεύρεσης στην μνήμη, ενώ πιο μεγάλες εφαρμογές με λεξιλόγια της τάξης των 100000 λέξεων κάνουν χρήση πιο έξυπνων γλωσσολόγων αποθηκεύοντας μόνον ορισμένο αριθμό δεδομένων από την διαδικασία εύρεσης στην μνήμη.

3.1.2.3 Ακουστικός βαθμολογητής. Ο στόχος του ακουστικού βαθμολογητή είναι να υπολογίσει την πιθανότητα της κατάστασης εξόδου ή τις τιμές πυκνότητας πιθανότητας για τις διάφορες καταστάσεις, για οποιοδήποτε δεδομένο διάνυσμα εισόδου. Αυτό γίνεται κατόπιν ζήτησης στην μονάδα αναζήτησης. Για να γίνει ο υπολογισμός των βαθμολογιών θα πρέπει

να υπάρξει επικοινωνία με το front end μπλοκ ώστε να αποκτήσει τα χαρακτηριστικά που έχουν εξαχθεί και στην συνέχεια να γίνει η βαθμολόγηση. Ο βαθμολογητής κρατάει όλες τις πληροφορίες σχετικά με τις πυκνότητες πιθανότητας των καταστάσεων εξόδου ενώ το μπλοκ αναζήτησης αγνοεί αν η βαθμολόγηση γίνεται με συνεχή, ημι-συνεχή ή διακριτά Κρυφά Μαρκοβιανά Μοντέλα (HMM). Τέλος η αρθρωτή φύση του σχεδιασμού μας επιτρέπει να συμπεριλάβουμε συμβατικές διαδικασίες βαθμολόγησης Gauss, καθώς και αριετούς νέους αλγορίθμους που βασίζονται στην θεωρία παιγνίων.

3.2 Αρχιτεκτονική λειτουργίας του Pocketsphinx

Η αρχιτεκτονική λειτουργίας του αποκωδικοποιητή του Pocketsphinx έχει προκύψει κυρίως από αυτήν του Sphinx-II.(Daines, 2011). Η γενική δομή της αρχιτεκτονικής δίνεται στο Σχεδιάγραμμα 6.



Σχεδιάγραμμα 6. Αρχιτεκτονική σχεδιασμού αποκωδικοποιητή Pocketsphinx
(Daines, 2011).

Το Pocketsphinx χρησιμοποιεί μια στρατηγική τριπλής αναζήτησης οι οποία έχει τα παρακάτω στάδια:

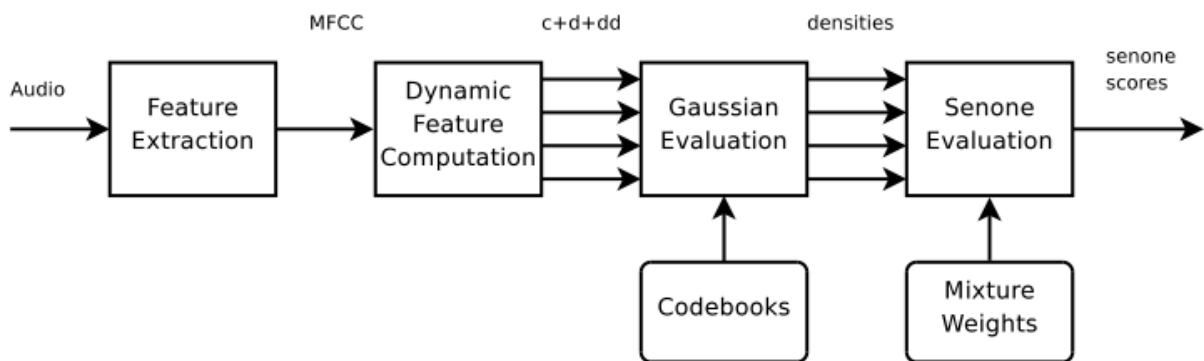
- FWDTREE Χρήση αλγόριθμου Viterbi και ενός στατικού δεντρικής μορφής λεξιλογίου.
- FWDFLAT Χρήση αλγόριθμου Viterbi και ενός επίπεδου λεξιλογίου.
- BESTPATH Εύρεση γραφήματος λέξεων.

Ο ίδιος ο αποκωδικοποιητής αποτελείται από τρία μέρη (μπλοκ):

- Ακουστική μοντελοποίηση.
- Αναζήτηση προς τα εμπρός.
- Εύρεση γραφήματος λέξεων.

3.2.1 Ακουστική μοντελοποίηση

Το Pocketsphinx έχει σχεδιαστεί ώστε να δουλεύει βέλτιστα με ημι-συνεχή ακουστικά μοντέλα (tied-mixture models) αν και μπορεί να υποστηρίξει και συνεχή μοντέλα (tied-state



Σχεδιάγραμμα 7. Ροή δεδομένων κατά την φάση της ακουστικής μοντελοποίησης (Daines, 2011).

models). Η χρήση ημι-συνεχών μοντέλων είναι πολύ σημαντική για την απόδοση του συστήματος, καθώς η δομή αυτών των μοντέλων μας επιτρέπει σαν έξοδο να έχουμε πυκνότητες πιθανοτήτων τύπου Gauss. Η ροή δεδομένων κατά την ακουστική μοντελοποίηση παρουσιάζεται στο Σχεδιάγραμμα 7. Ένα άλλο χαρακτηριστικό των ημι-συνεχών ακουστικών μοντέλων και της εργασίας μοντελοποίησης που λαμβάνει χώρα στο Pocketsphinx είναι ο χωρισμός του διανύσματος ακουστικού χαρακτηριστικού σε πολλαπλά μέρη, τα οποία μοντελοποιούνται το καθένα ξεχωριστά. Στα μέρη αυτά αποδίδονται μια σειρά από πυκνότητες πιθανοτήτων τύπου Gauss. Στην συνέχεια η ακουστική βαθμολογία μιας δεδομένης κατάστασης υπολογίζεται σαν το προϊόν των πιθανοτήτων που υπήρχαν στην πρότερη κατάσταση και του μοντέλου του μέρους αυτού. Αυτό μας δίνει την δυνατότητα να έχουμε μια μεγάλη γκάμα παραμέτρων εκπαίδευσης για να διαχειριστούμε έναν πολύ μικρό αριθμό Gauss αλλά και να κάνουμε πιο σωστούς υπολογισμούς των επιμέρους μερών κάτι που επιδρά άμεσα και στην χαμηλή χρήση της CPU του υπολογιστικού μας μηχανήματος.

3.2.2 Αποκωδικοποιητής

Κατά την φάση της αποκωδικοποίησης το Pocketsphinx χρησιμοποιεί μια προσέγγιση πολλαπλών αναζητήσεων, όπου κάθε αναζήτηση χρησιμεύει για να ξεπεράσει τις προσεγγίσεις που έγιναν από την προηγούμενη αναζήτηση. Όπως αναφέρθηκε και παραπάνω (3.2) τα τρία περάσματα (αναζητήσεις) κατά την λειτουργία του αποκωδικοποιητή του Pocketsphinx είναι γνωστά σαν fwdtree, fwdflat και bestpath και θα παρουσιαστούν στην συνέχεια ολοκληρώνοντας και την παρουσίαση της αρχιτεκτονικής λειτουργίας του Pocketsphinx.

3.2.2.1 Αναζήτηση κατά προσέγγιση με βάση δέντρο λέξεων (fwdtree). Η μέθοδος αναζήτησης fwdtree είναι βασισμένη σε ένα λεξικό το οποίο έχει μια δομή δεντρική. Χωρίζει το γράφημα αναζήτησης στο δέντρο του λεξιλογίου και στην αλληλουχία λέξεων. Για να έχουμε γρήγορη υπολογιστική διαδικασία η μέθοδος χρησιμοποιεί έναν μεγάλο αριθμό από ρυθμιζόμενες, κατά πλάτος, δέσμες. Συγκεκριμένα μια ξεχωριστή δέσμη εφαρμόζεται στο τελευταίο φώνημα κάθε λέξης από όλα τα Κρυφά Μαρκοβιανά Μοντέλα (HMM). Αυτό είναι εφικτό καθώς η βαθμολογία που έχουμε για το γλωσσικό μοντέλο έχει εφαρμοστεί στο τελευταίο φώνημα της λέξης καθιστώντας την βαθμολογία πιο διακριτή. Η μέθοδος αναζήτησης fwdtree είναι μεν γρήγορη αλλά όχι πολύ ακριβής.

3.2.2.2 Αναζήτηση κατά προσέγγιση με βάση ένα επίπεδο λεξιλόγιο (fwdflat). Η μέθοδος αναζήτησης fwdflat αναζητεί πάνω πλέγμα που έχει δημιουργήσει η προηγούμενη μέθοδος αναζήτησης (fwdtree) χωρίς όμως να βαθμολογήσει ξανά πάνω στο ίδιο πλέγμα. Ουσιαστικά βαθμολογεί ξανά λαμβάνοντας υπόψη της ότι το ενεργό λεξιλόγιο για αναζήτηση έχει ήδη υποτεθεί από την πρώτη ήδη αναζήτηση (fwdtree) και έτσι ο συνολικός αριθμός λέξεων σαν υπόθεση είναι σαφώς πιο περιορισμένος από τον αρχικό αριθμό λέξεων. Έτσι κατά την αποκωδικοποίηση έχουμε δυναμική συσχέτιση της κάθε φράσης μόνον με αυτό το σύνολο των λέξεων, κάτι που είναι βέλτιστο για φορητές συσκευές, όπου η υπολογιστική δύναμη και μνήμη είναι περιορισμένες.

3.2.2.3 Αναζήτηση με την δημιουργία πλέγματος (bestpath lattice generation). Η τρίτη μέθοδος αναζήτησης που χρησιμοποιεί το Pocketsphinx είναι ουσιαστικά μια αναβαθμολόγηση του πλέγματος λέξεων του πίνακα λέξεων που έχουν προκύψει από την αναζήτηση fwdflat, δημιουργώντας έτσι ένα νέο πλέγμα λέξεων. Η μεθοδολογία αναβαθμολόγησης στηρίζεται στον αλγόριθμο πιο σύντομης διαδρομής του Dijkstra (Dijkstra, 1959), όπου η βαθμολογίες των διαδρομών αναφέρονται σε τόξα και όχι κόμβους. Στην πραγματικότητα η πραγματοποίηση και των τριών μεθοδολογιών αναζήτησης είναι προαιρετικές, καθώς το Pocketsphinx μπορεί να υποθέσει για το ποια λέξη πρόκειται κάνοντας χρήση και μόνον από τα αποτελέσματα αναζήτησης της μεθόδου fwdtree. Τα αποτελέσματα

από τις άλλες δυο μεθοδολογίες αναζήτησης (fwdflat, bestpath) λειτουργούν διορθωτικά και συμπληρωματικά της πρώτης μεθοδολογίας.

Με την παρουσίαση των γενικών αρχών λειτουργίας του αποκωδικοποιητή του Pocketsphinx ολοκληρώθηκε η παρουσίαση των αρχιτεκτονικών λειτουργίας των CMU Sphinx και Pocketsphinx. Στο επόμενο Κεφάλαιο τρία (3) θα γίνει η αναλυτική παρουσίαση των εργασιών που εκτελέστηκαν στα πλαίσια αυτής της Μεταπτυχιακής διατριβής ώστε να δημιουργηθεί το Ελληνικό μοντέλο της πλατφόρμας CMU Sphinx.

ΚΕΦΑΛΑΙΟ 3

Δημιουργία Ελληνικού Μοντέλου για το CMU Sphinx

§1. Γενική περιγραφή του Ελληνικού μοντέλου για το CMU Sphinx

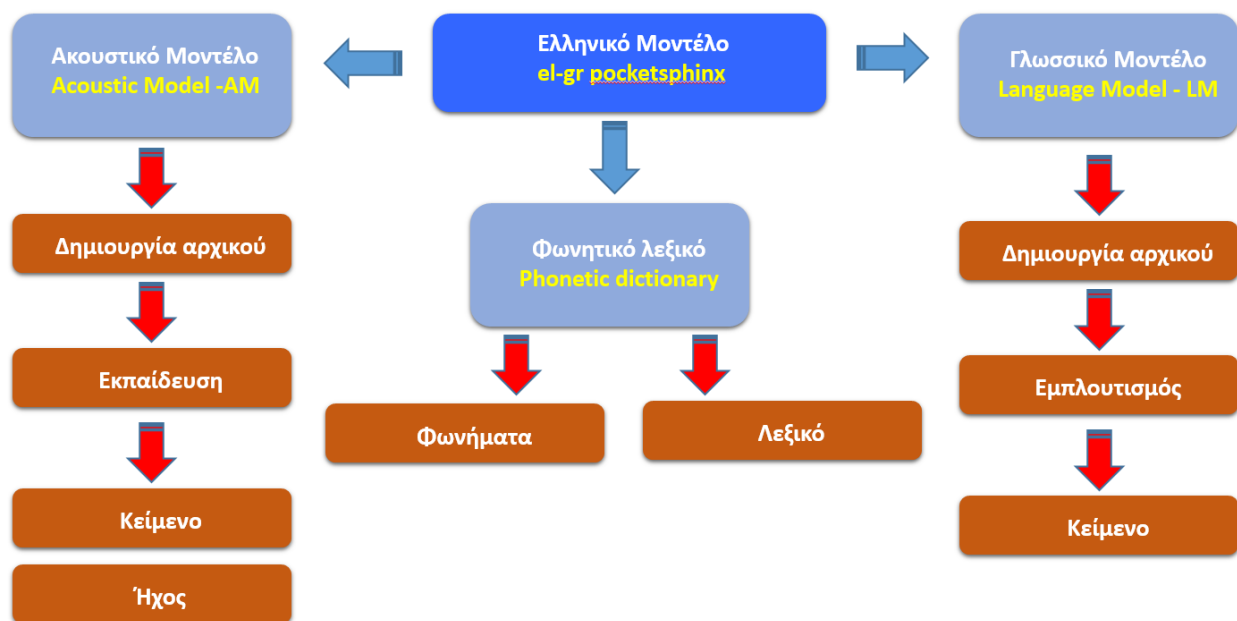
Η πλατφόρμα εργαλείων CMU Sphinx μας παρέχει μια σειρά από φωνητικά και γλωσσικών μοντέλων διαφόρων γλωσσών που αποτελούν και την βάση όλων των εργασιών που θέλουμε να κάνουμε κατά την διαδικασία της αυτόματης φωνητικής αναγνώρισης (ASR). Δημοφιλείς γλώσσες όπως τα Αγγλικά έχουν ήδη πολύ εμπεριστατωμένα και εκπαιδευμένα μοντέλα με αποτέλεσμα να έχουμε πολύ καλά ποσοστά επιτυχίας κατά την λειτουργία του συστήματος μας. Τα μοντέλα αυτά έχουν προκύψει από συνεχείς πειραματισμούς και εκπαιδεύσεις των μηχανισμών του CMU Sphinx και υποστηρίζονται από εκατοντάδες χρήστες που συμμετέχουν στην συνεχή βελτίωση τους.

Ένα από τα πολύ χρήσιμα χαρακτηριστικά του CMU Sphinx είναι το γεγονός ότι σου επιτρέπει να προσαρμόσεις την πλατφόρμα σύμφωνα με τις δικές σου ανάγκες. Καθώς το ηχόχρωμα φωνής αλλά και ο τρόπος ομιλίας του κάθε ατόμου διαφέρει, το σύστημα σου δίνει την δυνατότητα να το εκπαιδεύσεις, ώστε να ανταποκρίνεται με μεγαλύτερη ακρίβεια στις προσωπικές σου φωνητικές παραμέτρους. Άλλωστε αυτό είναι και η συνήθης πρακτική, ακόμη και όταν έχουμε να λειτουργήσουμε στην Αγγλική γλώσσα, όπου το αντίστοιχο μοντέλο είναι πολύ ακριβές.

Μέχρι πρότινος το Ελληνικό μοντέλο δεν υπήρχε στο διεθνές αποθετήριο του προγράμματος. Παρότι βέβαια ήδη ερευνητικές ομάδες είχαν εργαστεί για την δημιουργία του (Tsardoulis et al., 2015), αυτό περιοριζόταν σε συγκεκριμένο αριθμό λέξεων και δεν είχε την δυνατότητα να υποστηρίξει ένα μεγάλο λεξιλόγιο. Έτσι το πακέτο εργασιών που μπορούν να υποστηριχθούν είναι και αυτό ορισμένο από το λεξιλόγιο που υποστηρίζεται. Η φιλοσοφία ανάπτυξης ιδίως του φωνητικού λεξιλογίου έχει στηριχθεί στην μεθοδολογία που η παραπάνω ομάδα εργασίας εφάρμοσε κατά τις εργασίες της στο Ευρωπαϊκό ερευνητικό πρόγραμμα

RAPP (Robotic Applications for Delivering Smart User Empowering Applications) (<http://rapp-project.eu/>).

Ένα επιπλέον δεδομένο που υπήρχε, ήταν το γεγονός ότι δυστυχώς δεν είναι δυνατόν να προσαρμόσει κανείς, κάποιο ήδη έτοιμο μοντέλο ώστε να το χρησιμοποιήσει για την Ελληνική γλώσσα. Και αυτό έχει να κάνει με τις ιδιαιτερότητες και την μοναδικότητα της Ελληνικής γλώσσας. Συνεπώς η δημιουργία εκ του μηδενός του Ελληνικού μοντέλου ήταν αναγκαία. Το σύνολο των εργασιών που εκτελέστηκαν ώστε να είναι δυνατή η δημιουργία του Ελληνικού μοντέλου της πλατφόρμας CMU Sphinx, δίνεται στο Σχεδιάγραμμα 8.



Σχεδιάγραμμα 8. Πλάνο εργασιών δημιουργίας Ελληνικού μοντέλου για το CMU Sphinx

Βέβαια μετά την δημιουργία του Ελληνικού μοντέλου είναι δυνατόν κάποιος να το προσαρμόσει σύμφωνα με τις δικές του ανάγκες, όπως άλλωστε και όλα τα φωνητικά μοντέλα που η πλατφόρμα υποστηρίζει.

Όπως είναι κατανοητό μιλάμε για την δημιουργία του Ακουστικού μοντέλου, του Γλωσσικού μοντέλου και του Φωνητικού λεξικού. Στα επόμενα μέρη θα γίνει μια πιο αναλυτική περιγραφή των εργασιών δημιουργίας του Ελληνικού μοντέλου, ξεκινώντας από την δημιουργία του Γλωσσικού μοντέλου, στην συνέχεια την δημιουργία του Ελληνικού φωνητικού λεξικού και τέλος την δημιουργία του Ακουστικού μοντέλου που θα πρέπει να δημιουργηθεί και τελευταίο καθώς προϋποθέτει την ύπαρξη των δυο προηγούμενων.

§2. Διαδικασίες δημιουργίας του Ελληνικού Μοντέλου

2.1 Δημιουργία Ελληνικού Γλωσσικού μοντέλου (Language Model)

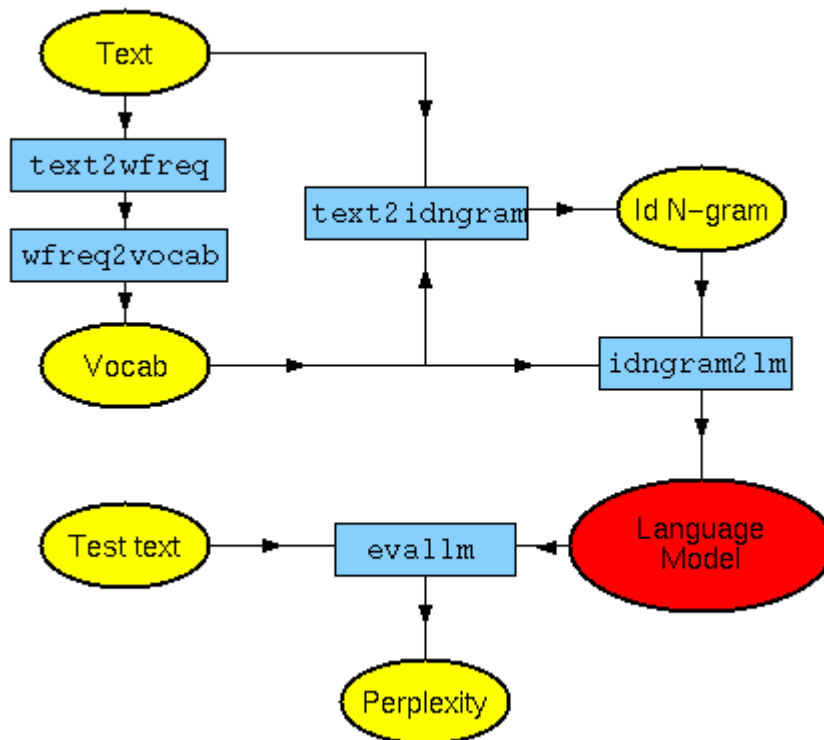
Όλες οι εργασίες για τις ανάγκες της Μεταπτυχιακής διατριβής εκτελέσθηκαν σε έναν υπολογιστή Dell Precision M6300 ο οποίος είχε επεξεργαστή Intel core Duo 4 gbite RAM και λειτουργικό σύστημα Linux Mint 18. Σε πρώτη φάση εγκαταστάθηκαν οι τελευταίες εκδόσεις των παρακάτω προγραμμάτων απαραίτητων για τις εργασίες sphinxbase, rocketsphinx, sphinxtrain μας. Στο διαδίκτυο υπάρχουν πολλές σελίδες και video με οδηγίες ώστε να αντιμετωπιστούν τα όποια μικρά προβλήματα κατά την φάση της εγκατάστασης, αλλά ίσως οι πιο ακριβείς οδηγίες δίνονται από τον Josh Meyer (<https://goo.gl/w6WK6f>) .

Το γλωσσικό μοντέλο (Language Model) υπολογίζει την πιθανότητα μιας υποθετικής ακολουθίας λέξεων, δημιουργώντας την βαθμολογία του γλωσσικού μοντέλου (LM score). Μπορεί έτσι να περιγράψει πιο πολύπλοκες γλώσσες και είναι ιδανικά για περιπτώσεις όπου έχουμε ελεύθερης μορφής, όπως ο προφορικός λόγος, εισόδους.

Υπάρχουν δυο τρόποι να δημιουργήσεις το γλωσσικό σου μοντέλο. Στην περίπτωση που έχεις ένα μικρό και περιορισμένο λεξιλόγιο είναι πιο εύκολο να κάνεις κανείς χρήση του online εργαλείου που έχει αναπτύξει το Πανεπιστήμιο του Carnegie Mellon (“Sphinx Knowledge Base Tool VERSION 3,” 2017). Στην περίπτωση όμως που θέλεις να δημιουργήσεις ένα μεγάλο γλωσσικό μοντέλο με περίπου 20000 λέξεις τότε θα πρέπει να ακολουθήσεις άλλη μεθοδολογία, τα βήματα της οποίας είναι :

- Εγκατάσταση του πακέτου προγραμμάτων CMUCLMTK (<https://goo.gl/xbE9MZ>)
- Προετοιμασία των αντίστοιχων κειμένων.
- Δημιουργία λίστας λέξεων.
- Δημιουργία γλωσσικού μοντέλου σύμφωνα με την μορφή που την δέχεται το σύστημα (.lm)
- Μετατροπή του γλωσσικού μοντέλου σε μορφή binary (.lm.bin)

Στην περίπτωση μας και σε πρώτη φάση έγινε η εγκατάσταση του πακέτου εργαλείων CMUCLMTK. Μια τυπική χρήση των εργαλείων που αυτό προσφέρει, δίνεται στην Εικόνα 23. Τα εργαλεία που χρησιμοποιήθηκαν είναι τα: text2wfreq, wfreq2vocab, text2idngram, idngram2lm.



Εικόνα 23. Τυπική χρήση εργαλείων δημιουργίας Γλωσσικού μοντέλου
(<http://www.speech.cs.cmu.edu>)

text2wfreq : κατασκευάζει μια λίστα με όλες τις λέξεις που έχουμε καθώς και με έναν αριθμό που δείχνει τις συνολικές εμφανίσεις της κάθε λέξης.

wfreq2vocab : παίρνει σαν είσοδο το αρχείο που παράγει το text2wfreq και μας δίνει σαν έξοδο ένα λεξιλόγιο σε μορφή ASCII. Το λεξιλόγιο περιορίζεται σε 65535 λέξεις ενώ μπορούν επίσης να συμπεριληφθούν τα σχόλια (κάθε γραμμή που αρχίζει ## θεωρείται σχόλιο).

text2idngram : Δέχεται σαν είσοδο το λεξιλόγιο και μια ροή κειμένου και δίνει σαν έξοδο μια λίστα με όλους τους αύξοντες αριθμούς n-grams αριθμών που αντιστοιχούν στις χαρτογραφήσεις των λέξεων.

idngram2lm : Παίρνει την είσοδο από το text2idngram και μας δίνει το Γλωσσικό μοντέλο

Το επόμενο στάδιο ήταν η προετοιμασία των κειμένων που θα τροφοδοτούσαν τα εργαλεία του πακέτου CMUCLMTK ώστε να έχουμε την παραγωγή του Ελληνικού γλωσσικού μοντέλου.

Τα κείμενα μας καταρχήν θα πρέπει να έχουν συγκεκριμένη μορφή. Δεν θα πρέπει να υπάρχουν κεφαλαία γράμματα κενά και σημεία στίξης πέραν των τόνων. Οι ξεχωριστές φράσεις θα πρέπει να οροθετούνται με τα σύμβολα <s> και </s> ενώ δεν θα πρέπει να υπάρχουν σύμβολα αριθμών. Οι αριθμοί όμως μπορούν να υπάρχουν σε λεκτική μορφή. Προφανώς όσο πιο πολλά κείμενα, διαφορετικής θεματολογίας χρησιμοποιήσουμε, τόσο μεγαλύτερη ακρίβεια θα έχουμε στο λεκτικό μας μοντέλο.

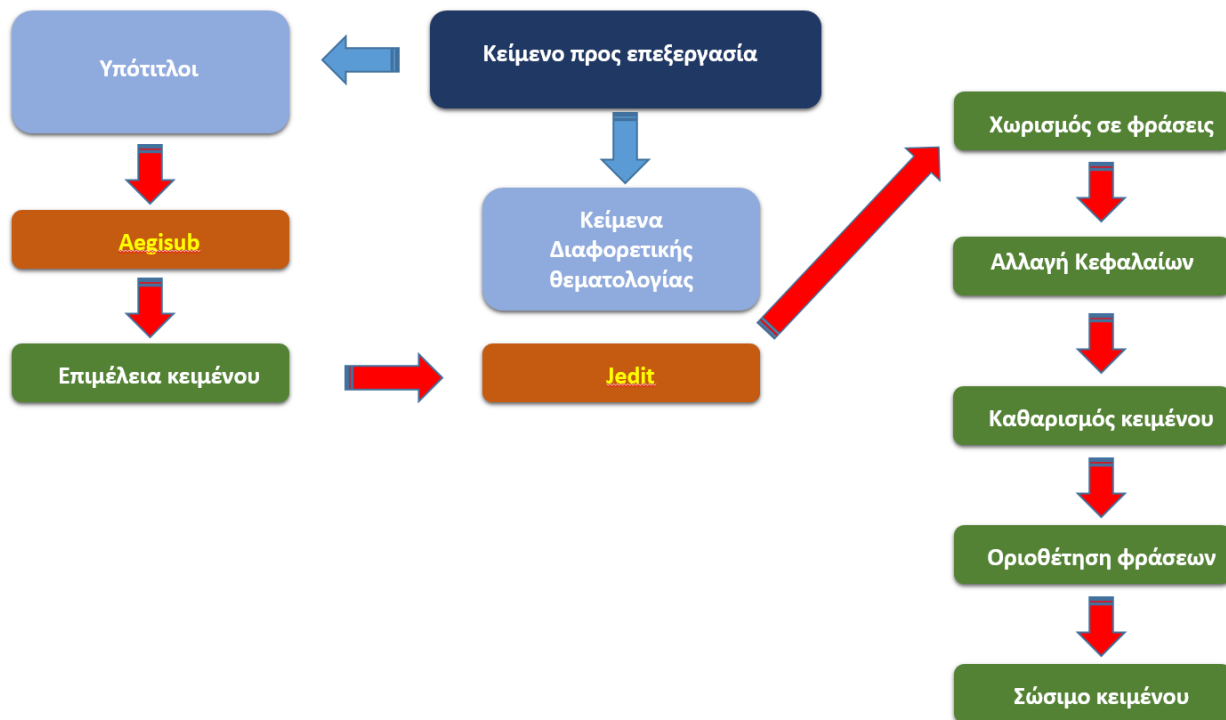
Σαν κείμενα δημιουργίας χρησιμοποιήθηκαν κείμενα από την καθημερινή ειδησεογραφία, ελεύθερα κείμενα από βιβλία διαφορετικής θεματολογίας, καθώς και υπότιτλοι από ντοκιμαντέρ. Κύρια κριτήρια κατά την επιλογή των κειμένων υπήρξε αυτά να ανταποκρίνονται όσο τον δυνατόν στην απλή καθομιλούμενη γλώσσα και να μην περιέχουν τεχνικούς όρους ή εξειδικευμένο λεξιλόγιο.

Ιδίως για τους υπότιτλους αυτοί πρώτα επεξεργάστηκαν με το πρόγραμμα ανοιχτού κώδικα Aegisub (“Aegisub Advanced Subtitle Editor,” 2017) ώστε να έρθουν σε μια μορφή επεξεργασίας που να μας επιτρέπει να τα τροφοδοτήσουμε στο πακέτο CMUCLMTK.

Όλα τα κείμενα επεξεργάστηκαν και επιμελήθηκαν με το πρόγραμμα ανοιχτού κώδικα Jedit (“jEdit - Programmer’s Text Editor,” 2017) το οποίο αποδείχθηκε πολύ εύχρηστο, ιδίως στο προγραμματισμό μακροεντολών που αυτοματοποίησαν αρκετά την όλη διαδικασία. Επίσης τα ίδιο προσφέρει μια σειρά δικών του μακροεντολών που βοηθήσαν πολύ στις εκτελούμενες εργασίες.

Θα πρέπει εδώ να αναφερθεί ότι το γλωσσικό μοντέλο είναι άρρηκτα συνδεδεμένο με το φωνητικό λεξιλόγιο. Έτσι για παράδειγμα αν υπάρχει μια λέξη στο γλωσσικό μοντέλο που δεν υπάρχει στο φωνητικό λεξιλόγιο, το Pocketsphinx δεν μπορεί να προχωρήσει στην διαδικασία της εκπαίδευσης του. Το πρόβλημα ευτυχώς επιλύεται εύκολα με την προσθήκη της λέξης αυτής, στο φωνητικό λεξιλόγιο αφού βέβαια την μετατρέψουμε στο αντίστοιχο format. Επίσης θα πρέπει να αναφερθεί ότι το γλωσσικό μοντέλο μπορεί να εμπλουτίζεται συνεχώς με την προσθήκη επιπλέον κειμένων. Αυτή την στιγμή το Ελληνικό γλωσσικό μοντέλο αποτελείται από παραπάνω από 19000 λέξεις και η ροή των εργασιών δίνεται παρακάτω στο Σχεδιάγραμμα 9, ενώ ο κώδικας των μακροεντολών που χρησιμοποιήθηκαν για την επιμέλεια

των κειμένων ([breakLines](#), [CleanEmAll](#)) δίνονται στο παράρτημα της Μεταπτυχιακής διατριβής.



Σχεδιάγραμμα 9. Ροή εργασιών κατά την δημιουργία του Γλωσσικού μοντέλου

Έτσι για παράδειγμα αν έχουμε το παρακάτω κείμενο :

“Φρένο στη δράση της ηλεκτρονικής πλατφόρμας Uber για τη μίσθωση όχι μόνον ταξί, αλλά και ΙΧ, για επιβατικές μεταφορές εντός της ΕΕ βάζει πρόταση-εισήγηση του ευρωπαίου γενικού εισαγγελέα προς το Ευρωπαϊκό Δικαστήριο. Κατά τον γενικό εισαγγελέα, η πλατφόρμα Uber δεν διέπεται από την αρχή της ελεύθερης παροχής ψηφιακών υπηρεσιών (της κοινωνίας της πληροφορίας), αλλά η δραστηριότητά της εμπίπτει στις δεσμεύσεις τομέα των μεταφορών όπου η κατοχή αδείας μεταφοράς είναι υποχρεωτική. Ο εισαγγελέας επισημαίνει, όπως αναφέρουν Τα Νέα, ότι δεν ισχύει υπέρ της Uber η αρχή της ελεύθερης παροχής υπηρεσιών την οποία διασφαλίζει το δικαίο της Ένωσης για τις υπηρεσίες της κοινωνίας της πληροφορίας, θεωρώντας ότι δεν είναι δυνατό να χαρακτηριστεί ως τέτοια.”

Μετά την επεξεργασία που κάνουμε ώστε να είναι έτοιμο για το πακέτο CMUCLMTK αυτό γίνεται :

<s> φρένο στη δράση της ηλεκτρονικής πλατφόρμας uber για τη μίσθωση όχι μόνον ταξί αλλά και ιχ για επιβατικές μεταφορές εντός της εε βάζει πρόταση εισήγηση του ευρωπαϊού γενικού εισαγγελέα προς το ευρωπαϊκό δικαστήριο </s>

<s> κατὰ τον γενικό εισαγγελέα η πλατφόρμα uber δεν διέπεται από την αρχή της ελεύθερης παροχής ψηφιακών υπηρεσιών της κοινωνίας της πληροφορίας αλλά η δραστηριότητά της εμπίπτει στις δεσμεύσεις τομέα των μεταφορών όπου η κατοχή αδείας μεταφοράς είναι υποχρεωτική </s>

<s> ο εισαγγελέας επισημαίνει όπως αναφέρουν τα νέα ότι δεν ισχύει υπέρ της uber η αρχή της ελεύθερης παροχής υπηρεσιών την οποία διασφαλίζει το δίκαιο της ένωσης για τις υπηρεσίες της κοινωνίας της πληροφορίας θεωρώντας ότι δεν είναι δυνατό να χαρακτηριστεί ως τέτοια </s>

Όταν τροφοδοτήσουμε το πακέτο CMUCLMTK με την παρακάτω αλληλουχία εντολών :

```
text2wfreq < text.txt | wfreq2vocab > text.vocab
text2idngram -vocab text.vocab -idngram text.idngram < text.txt
idngram2lm -vocab_type 0 -idngram text.idngram -vocab \
text.vocab -arpa test.lm
sphinx_lm_convert -i test.lm -o test.lm.bin
```

θα έχουμε την παραγωγή του Ενδεικτικού γλωσσικού μοντέλου [test.lm](#) (παρατίθεται στο Παράρτημα) το οποίο αν θέλουμε το μετατρέπουμε σε μορφή binary ώστε να είναι πιο γρήγορη η υπολογιστική του διαχείριση.

Μετά την δημιουργία του αρχείου με τα κείμενα, που θα μας δημιουργούσαν το γλωσσικό μοντέλο, έγινε η τροφοδοσία του στο CMUCLMTK και στην συνέχεια είχαμε την δημιουργία του el-gr.lm.bin που αποτελεί και το γλωσσικό μοντέλο του συνολικού Ελληνικού μοντέλου. Προφανώς με τον ίδιο τρόπο αν θέλουμε να επικεντρωθούμε σε συγκεκριμένη θεματολογία μπορούμε να δημιουργήσουμε γλωσσικά μοντέλα που να έχουν λέξεις μόνον από αυτή την ενότητα, αυξάνοντας την ακρίβεια του συστήματος μας σε προφορικό λόγο που προέρχεται, αποκλειστικά μόνον, από τον συγκεκριμένο θεματικό χώρο. Αυτό και έγινε στο μικρότερο και επικεντρωμένο σε εργασίες με ROV λεκτικό μοντέλο.

2.2 Δημιουργία Ελληνικού φωνητικού λεξικού (Phonetic dictionary)

Ένα από τα πιο σημαντικά μέρη του Ελληνικού μοντέλου είναι το Ελληνικό φωνητικό λεξικό (Phonetic dictionary). Πρόκειται ουσιαστικά για την μετατροπή των Ελληνικών λέξεων σε μια μορφή η οποία αν είναι κατανοητή από το σύστημα λειτουργίας του CMU Sphinx και κατά συνέπεια του Pocketsphinx.

Για να μπορέσουμε να μετατρέψουμε τις λέξεις (στην περίπτωση μας Ελληνικές) θα χρειαστούμε σε πρώτη φάση ένα συγκεκριμένο αλφάβητο. Το αλφάβητο που χρησιμοποιείται διεθνώς για αυτές τις εργασίες στην πλατφόρμα CMU Sphinx είναι το ARPABET. Το ARPABET είναι η φωνητική αναπαράσταση που αναπτύχθηκε από την ARPA (Advanced Research Projects Agency) στα πλαίσια προγραμμάτων κατανόησης του λόγου. Κάθε φώνημα του Αγγλικού αλφαβήτου έχει αντιστοιχιστεί με μια ξεχωριστή αλληλουχία χαρακτήρων ASCII. Έτσι για την Αγγλική γλώσσα που έχει 39 φωνήματα ένα παράδειγμα μετατροπής λέξεων σε μορφή κατανοητή από την πλατφόρμα δίνεται στην Εικόνα 24.

Γίνεται συνεπώς κατανοητό ότι για την δημιουργία του Ελληνικού φωνητικού λεξικού θα πρέπει να κάνουμε δυο επιμέρους εργασίες. Σε πρώτη φάση να βρούμε έναν ικανό αριθμό Ελληνικών λέξεων και να δημιουργήσουμε το λεξιλόγιο μας, και σε δεύτερη φάση να δημιουργήσουμε τον μηχανισμό μετατροπής αυτών των λέξεων σε μορφή κατανοητή από το σύστημα.

Phoneme	Example	Translation
AA	odd	AA D
AE	at	AE T
AH	hut	HH AH T
AO	ought	AO T
AW	cow	K AW
AY	hide	HH AY D
B	be	B IY
CH	cheese	CH IY Z
D	dee	D IY
DH	thee	DH IY
EH	Ed	EH D
ER	hurt	HH ER T
EY	ate	EY T
F	fee	F IY
G	green	G R IY N
HH	he	HH IY
IH	it	IH T
IY	eat	IY T
JH	gee	JH IY
K	key	K IY
L	lee	L IY
M	me	M IY
N	knee	N IY
NG	ping	P IH NG
OW	oat	OW T
OY	toy	T OY
P	pee	P IY
R	read	R IY D
S	sea	S IY
SH	she	SH IY
T	tea	T IY
TH	theta	TH EY T AH
UH	hood	HH UH D
UW	two	T UW
V	vee	V IY
W	we	W IY
Y	yield	Y IY L D
Z	zee	Z IY
ZH	seizure	S IY ZH ER

Εικόνα 24. Παράδειγμα μετατροπής λέξεων με την χρήση του ARPABET
(<http://www.speech.cs.cmu.edu>)

2.2.1 Δημιουργία λεξιλογίου

Για την δημιουργία του λεξιλογίου προτιμήθηκε η χρήση λεξιλογίων που είναι μέρη προγραμμάτων ανοιχτού κώδικα. Θα πρέπει να τονιστεί εδώ, ότι κανένα τυποποιημένο λεξιλόγιο δεν μπορεί να έχει συμπεριλάβει όλες τις λέξεις που είναι δυνατόν να συναντήσουμε στην καθημερινότητα μας. Έτσι για παράδειγμα επώνυμα ή τοπωνύμια είναι δυνατόν να μην τα συναντήσουμε σε τυποποιημένα λεξιλόγια. Πλην όμως καθώς αυτά χρησιμοποιούνται από μας θα πρέπει να τα προσθέσουμε στο συνολικό φωνητικό μας λεξιλόγιο ώστε αυτό να ανταποκρίνεται όσο το δυνατόν καλύτερα στην πραγματικότητα.

Σαν μια πρώτη πηγή λέξεων χρησιμοποιήθηκε το αρχείο λέξεων που προσφέρεται από το Ινστιτούτο Επεξεργασίας Λόγου και αποτελεί αποτέλεσμα δουλειάς των ερευνητών του (Protopapas, Tzakosta, Chalamandaris, & Tsiakoulis, 2012) .

Σαν δεύτερη πηγή λέξεων χρησιμοποιήθηκε το Ελληνικό λεξικό για το πρόγραμμα OpenOffice/LibreOffice (“Ελληνικός ορθογραφικός και γραμματικός έλεγχος,” 2017) όπως αυτό είχε διαμορφωθεί από τους δημιουργούς του κατά την περίοδο που λάμβαναν χώρα οι εργασίες δημιουργίας του φωνητικού λεξικού (2017).

Τέλος έχουν προστεθεί λέξεις που προέκυψαν κατά την φάση δημιουργίας του γλωσσικού μοντέλου και δεν υπήρχαν στα προαναφερθέντα λεξικά. Οι λέξεις επεξεργάστηκαν ώστε να μην υπάρχουν διπλές εγγραφές και στην συνέχεια έγινε η διαδικασία μετατροπής των με την βοήθεια του ARPABET και συγκεκριμένων κανόνων στην μορφή που επιθυμούσαμε για χρήση στην πλατφόρμα CMU Sphinx.

2.2.2 Δημιουργία μηχανισμού μετατροπής λέξεων σε σύνολο φωνημάτων

Μετά την δημιουργία του λεξικού έπρεπε να δημιουργηθεί ο μηχανισμός που θα μετέτρεπε τις λέξεις αυτές, σε ένα σύνολο φωνημάτων κατανοητό από το σύστημα του CMU Sphinx. Συνήθως αυτή η εργασία γίνεται με εξειδικευμένα εργαλεία όπως το [phonetisaurus](#) ή το [sequitur-g2p3](#) (Bisani & Ney, 2008). Εδώ όμως προτιμήθηκε να γίνει η μετατροπή κατευθείαν σε μορφή κατανοητή από το CMU Sphinx. Άρα συνεπώς έπρεπε να καθοριστούν τα Ελληνικά φωνήματα και στην συνέχεια να ορισθούν τα διαδοχικά βήματα που έπρεπε να ακολουθηθούν ώστε να μπορούμε να κάνουμε την μετατροπή. Για την διαδικασία της μετατροπής προτιμήθηκε πάλι το ελεύθερο πρόγραμμα Jedit (“jEdit - Programmer’s Text Editor,” n.d.) όπου και δημιουργήθηκε η αντίστοιχη μακροεντολή [create_phoneme](#) που παρατίθεται στο παράρτημα.

Στην Ελληνική γλώσσα έχουμε ως γνωστόν 24 γράμματα (α, β, γ, δ, ε, ζ, η, θ, ι, κ, λ, μ, ν, ξ, ο, π, ρ, σ, τ, υ, φ, χ, ψ, ω) και 25 φωνήματα (α, ε, ι, ο, ου, β, γ, δ, ζ, θ, κ, λ, μ, ν, π, ρ, σ, τ, φ, χ, μπ, ντ, γκ, τσ, τζ) ενώ το Ελληνικό αλφάβητο δεν έχει ξεχωριστά γράμματα για κάθε φώνημα. Υπάρχουν φωνήματα που περιέχουν δυο γράμματα (πχ. γκ) ενώ υπάρχουν και ομόηχα γράμματα όπως για παράδειγμα τα γράμματα (η, ι, υ) που αντιστοιχούν φωνητικά στο φώνημα ι. Υπάρχουν τα διπλά γράμματα (ξ, ψ) ενώ φυσικά υπάρχουν και τα διπλά γράμματα τα οποία αντιστοιχούν σε ένα μονό φώνημα όπως το αι που αντιστοιχεί στο ε. Στο Ελληνικό αλφάβητο έχουμε το τελικό σίγμα ενώ έχουμε και τα διπλά γράμματα όπως τα ββ, κκ, λλ κτλ. Τέλος υπάρχουν και τα αυ και ευ που αναλόγως με ο τι γράμμα ακολουθεί έχουν και διαφορετική φωνητική απεικόνιση (Tsardoulis et al., 2015).

Λαμβάνοντας υπόψη όλα τα παραπάνω και τα φωνήματα που το CMU Sphinx χρησιμοποιεί καταλήγουμε στον παρακάτω κατάλογο φωνημάτων που χρησιμοποιούμε στο Ελληνικό μοντέλο.

Κατάλογος φωνημάτων Ελληνικού μοντέλου

ΑΕ, ΑΑ, Β, CH, D, DH, EH, F, G, HH, IH, JH, K, KS, L, M, N, OW, P, PS, R, S, SIL, T, TH, UW, V, W, Z, ZB, ZD, ZDH, ZL, ZM, ZN, ZR, ZV, ZW

Έχοντας τα φωνήματα και τις λέξεις στην συνέχεια καθορίστηκε η σειρά του αλγόριθμου μετατροπής και αυτός εφαρμόστηκε σαν μακροεντολή στο Jedit. Τα βήματα της μακροεντολής [create_phoneme](#) (παρατίθεται στο παράρτημα) για την μετατροπή μια λέξης σε μορφή ARPABET είναι:

1. Μετατροπή όλων των γραμμάτων σε μικρά.
2. Αντικατάσταση των διπλών και τριπλών συνδυασμών γραμμάτων με τα αντίστοιχα φωνήματα, με την παρακάτω σειρά: ου, ού, σμπ, σντ, σγ, σβ, σδ, σμ, σν, σλ, σρ, μπ, ντ, γκ, γγ, τς, τζ, σς, κκ, ββ, λλ, μμ, νν, ππ, ρρ, ττ
3. Αντικατάσταση των διπλών ειδικών συνδυασμών φωνηέντων με τα αντίστοιχα φωνήματα, με την παρακάτω σειρά: αι, αί, ει, εί, οι, οί, υι, υί, αυ, αβ, ευ, εβ, εύ, έβ, αυ, αφ, αύ, άφ, έφ
4. Αντικατάσταση όλων των υπολοίπων γραμμάτων με τα αντίστοιχα φωνήματα.

Μερικά παραδείγματα μετατροπής λέξεων δίνονται παρακάτω :

Μερικά	M E H R I H K A A
παραδείγματα	P A A R A A D H I H W M A A T A A
μετατροπής	M E H T A A T R O W P I H S
λέξεων	L E H K S E H O W N
δίνονται	D H I H N O W D E H
στο	S T O W
παρακάτω	P A A R A A K A A T O W
πίνακα	P I H N A A K A A

Για την ολοκλήρωση του Ελληνικού φωνητικού λεξικού στην συνέχεια έγινε εφαρμογή του συγκεκριμένου αλγόριθμου μετατροπής στο ήδη δημιουργημένο λεξικό από την πρώτη φάση της εργασίας. Μετά και την ολοκλήρωση του φωνητικού λεξικού οι εργασίες επικεντρώθηκαν στην εκπαίδευσή του Ελληνικού ακουστικού μοντέλου, εργασίες που θα παρουσιασθούν στην συνέχεια.

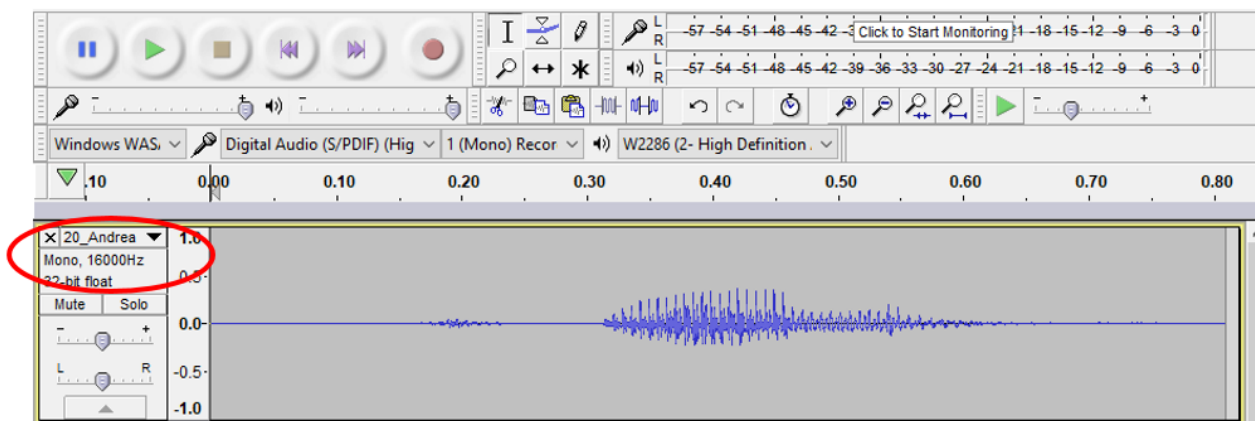
2.3 Δημιουργία Ελληνικού Ακουστικού μοντέλου (Acoustic Model)

Το τρίτο, πολύ σημαντικό, και πιο δύσκολο να δημιουργηθεί κομμάτι του Ελληνικού μοντέλου, ήταν το Ακουστικό μοντέλο (Acoustic Model). Πρόκειται ουσιαστικά για μια σειρά στατιστικών αρχείων που περιέχουν πληροφορίες σχετικά με μονάδες του λόγου. Είναι ουσιαστικά η «χαρτογράφηση» της κάθε γλώσσας που θέλουμε να χρησιμοποιήσουμε σε υπηρεσίες φωνητικής αναγνώρισης. Για να προέλθει αυτή η χαρτογράφηση θα πρέπει να τροφοδοτήσουμε το σύστημα μας με έναν μεγάλο αριθμό ακουστικών αρχείων τα οποία το σύστημα να εξετάζει σε αντιδιαστολή με γραπτό κείμενο.

Εκεί έγκειται και η δυσκολία αυτού του μέρους. Χρειάζεται μεγάλος όγκος δεδομένων και αρκετές μέρες εργασίας του υπολογιστή ώστε να εκπαιδευτεί σωστά το ακουστικό μοντέλο. Για να μπορέσουμε να επιτύχουμε όσο το δυνατό πιο επιτυχημένο ακουστικό μοντέλο θα πρέπει καταρχήν να έχουμε πληθώρα διαφόρων ηχητικών σημάτων από διαφορετικούς ομιλητές και ομιλήτριες και προφανώς να έχουμε ξεχωριστή θεματολογία λόγου. Δεν θέλουμε για παράδειγμα να τροφοδοτήσουμε το σύστημα μας με ηχητικά δεδομένα που να

αναφέρονται μόνον σε στρατιωτικά ή οικονομικά θέματα, καθώς στην συνέχεια θα πετύχουμε πολύ μικρές ακρίβειες στην αναγνώριση του λόγου.

Τα ηχητικά δεδομένα δε, πρέπει να έχουν συγκεκριμένη κωδικοποίηση (format) ώστε να γίνουν αποδεκτά από το Pocketsphinx και το sphinxtrain που είναι το πρόγραμμα του πακέτου που αναλαμβάνει την εκπαίδευση του ακουστικού μοντέλου. Συγκεκριμένα τα ηχητικά μας δεδομένα θα πρέπει να είναι σωσμένα σε μορφή MS WAV, μονοφωνικά (mono) στα 16 kHz 16 bit όπως φαίνεται και στην Εικόνα 25.



Εικόνα 25. Δείγμα ηχητικού σήματος που χρησιμοποιήθηκε στην διαδικασία εκπαίδευσης του ακουστικού μοντέλου

Η εκπαίδευσή του Ελληνικού ακουστικού μοντέλου στηρίχθηκε στην μεθοδολογία εκπαίδευσης που αναπτύχθηκε από την ομάδα εργασίας του Carnegie Mellon University (<https://cmusphinx.github.io/wiki/tutorialam/>) τα βασικά βήματα τις οποίας είναι :

1. Εγκατάσταση απαραίτητου λογισμικού (sphinxtrain, training base).
2. Προετοιμασία ηχητικού υλικού εκπαίδευσης.
3. Παραμετροποίηση μηχανισμού εκπαίδευσης.
4. Τροφοδοσία sphinxtrain με υλικό εκπαίδευσης.

2.3.1 Εγκατάσταση απαραίτητου λογισμικού (sphinxtrain, training base)

Το πρώτο στάδιο των εργασιών ήταν και το πιο απλό, καθώς απλά έγινε η εγκατάσταση των κατάλληλων λογισμικών ώστε να είναι δυνατή η διαδικασία της εκπαίδευσης. Έτσι εγκαταστάθηκε το sphinxtrain μέρος του πακέτου του CMU Sphinx αλλά και η «βάση εκπαίδευσης» [an4](#) (AlphaNumeric4) που δίνεται από το Carnegie Mellon University. Ο

φάκελος an4 είναι και ο χώρος εργασίας όπου θα γίνει η όλη διαδικασία και όπου θα αποθηκευτούν όλα τα στατιστικά αρχεία που θα αποτελέσουν το Ελληνικό ακουστικό μοντέλο.

2.3.2 Προετοιμασία ηχητικού υλικού εκπαίδευσης

Για να μπορέσουμε να τροφοδοτήσουμε το σύστημα μας με ηχητικά δεδομένα αυτά θα πρέπει να είναι σωσμένα σε μορφή MS WAV, μονοφωνικά (mono) στα 16 kHz και 16 bit. Για να είναι δυνατόν η όσο το δυνατόν πιο αντιπροσωπευτική εκπαίδευση του μοντέλου επιλέχθηκαν μια σειρά από διαφορετικές πηγές ήχου, ώστε να επιτευχθεί πολυφωνία θεματική και ηχητική. Έτσι επιλέχθηκαν:

- **Ηχογραφήσεις φυσικών προσώπων.** Ζητήθηκε από φίλους και φίλες να διαβάσουν ένα κείμενο 30 προτάσεων, οι οποίες δημιουργήθηκαν με το σκεπτικό να περιέχουν μέσα τους λέξεις ελέγχου προς ένα αυτόνομο όχημα. [Το κείμενο](#) παρατίθεται στο παράρτημα.
- **Ηχογράφηση ειδησεογραφίας.** Επιλέχθηκαν κείμενα της τρέχουσας ειδησεογραφίας, από διάφορες θεματικές ενότητες, τα οποία στην συνέχεια με την βοήθεια της υπηρεσίας Google translate (<https://translate.google.com/>) εκφωνήθηκαν σε συνθετική γυναικεία γλώσσα και ηχογραφήθηκαν την ίδια στιγμή με την χρήση του προγράμματος ελεύθερου λογισμικού Audacity (<http://www.audacityteam.org/>).
- **Ομιλούντα βιβλία.** Χρησιμοποιήθηκαν αποσπάσματα από ομιλούντα βιβλία του Υπουργείου Παιδείας, όπως αυτά είναι διαθέσιμα στην αντίστοιχη ιστοσελίδα του υπουργείου (“Διαδραστικά Σχολικά Βιβλία,” 2017).
- **Ηχητικά δεδομένα διαλέξεων.** Μια πολύ σημαντική πηγή καλών ηχητικών δεδομένων αποτέλεσαν οι ανοιχτές διαλέξεις Ελληνικών πανεπιστημίων που βρίσκονται στο αποθετήριο του προγράμματος Open courses (“Open Courses,” n.d.), αλλά και οι διαλέξεις των καθηγητών μου, από τα μαθήματα του δια τμηματικού μεταπτυχιακού προγράμματος σπουδών που παρακολούθησα (<http://sse-tuc.edu.gr/>). Οι διαλέξεις που στην πλειοψηφία τους είναι οπτικό ακουστικό υλικό επεξεργάστηκαν ώστε να υπάρξει εξαγωγή μόνον του ηχητικού μέρους και μάλιστα στην μορφή που το θέλουμε για την εκπαίδευση.

- **Συνεδριάσεις Ελληνικού κοινοβουλίου.** Το αποθετήριο οπτικοακουστικού υλικού του Ελληνικού κοινοβουλίου διαθέτει πληθώρα υλικού που χαρακτηρίζεται από διαφορετική θεματολογία αλλά και πολλούς και διαφορετικούς ομιλητές και ομιλήτριες. (<http://www.parliament.gr/>) Και εδώ το υλικό αφού επεξεργάστηκε σώθηκε στην κατάλληλη μορφή ώστε να είναι δυνατή η τροφοδοσία του στο sphinxtrain.

Συνολικά το ηχητικό υλικό που χρησιμοποιήθηκε για την εκπαίδευση στο sphinxtrain είχε μια συνολική διάρκεια **358:09:38 ωρών**.

2.3.3 Παραμετροποίηση μηχανισμού εκπαίδευσης

Μετά την προετοιμασία του ηχητικού υλικού, έγινε η παραμετροποίηση του μηχανισμού εκπαίδευσης.

Συγκεκριμένα, όλα τα ηχητικά μέρη, αριθμήθηκαν (κατά αύξουσα σειρά), μετονομάστηκαν (για ευκολία χρήσης) ενώ στην συνέχεια τοποθετήθηκαν στον φάκελο της βάσης εκπαίδευσης (an4). Ο λόγος που έγινε η κατά αύξουσα σειρά αρίθμηση έγκειται στο γεγονός της προετοιμασίας των αντίστοιχων αρχείων μεταγραφής. Πρόκειται για δυο αρχεία σε μορφή κειμένου txt.

Το πρώτο έχει, ανά σειρά, τα κείμενα που αντιστοιχούν στα αρχεία ήχου κατά απόλυτη ταύτιση. Το κείμενο έχει την παρακάτω μορφή :

```
<s> ξεινά πορεία ευθεία στρίψε δεξιά </s> (108)
<s> σταμάτα ξεινά στρίψε αριστερά ξεινά ξανά </s> (109)
<s> κάνε στροφή εκατόν ογδόντα μοιρών </s> (110)
```

Τα σύμβολα <s>, </s> οριοθετούν την έναρξη και το τέλος της φράσης που ακούγεται στο ηχητικό αρχείο, και το (108) δίνει το όνομα του ηχητικού αρχείου. Προφανώς αν είχαμε ονομάσει το ηχητικό μας αρχείο istoria.wav θα είχαμε αντίστοιχα:

```
<s> ξεινά πορεία ευθεία στρίψε δεξιά </s> (istoria).
```

Το δεύτερο αρχείο, που πρέπει να ετοιμάσουμε είναι αυτό που θα έχει σε απόλυτη κατάταξη όλα τα ονόματα των ηχητικών αρχείων. Για το κείμενο που χρησιμοποιήσαμε παραπάνω θα είχε την μορφή :

108

109

110

Αν τα αρχεία μας ονομαζόταν `alpha.wav`, `bravo.wav`, `charlie.wav` ο κατάλογος μας αντίστοιχα θα ήταν :

`alpha`

`bravo`

`charlie`

Τα δυο αυτά πολύ σημαντικά text αρχεία ονομάζονται αντίστοιχα `train.transcription` και `train.fileids`. Αν θέλουμε στο τέλος της εκπαίδευσης να τεστάρουμε την ακρίβεια του μοντέλου μας, δεν έχουμε παρά να ετοιμάσουμε ένα πακέτο δοκιμής που αντίστοιχα θα περιλαμβάνει :

- Ηχητικά αρχεία για να δοκιμάσουμε το μοντέλο.
- Το αρχείο `test.transcription`, που είναι ακριβώς στην ίδια φιλοσοφία με το `train.transcription`.
- Το αρχείο `test.fileids`, που είναι ακριβώς στην ίδια φιλοσοφία με το `train.fileids`.

Είναι εύκολα κατανοητό ότι χρειάστηκαν πολλές εργατοώρες ώστε να υπάρξει η πιο ακριβής μεταφορά σε κείμενο των ηχητικών σημάτων, ώστε να επιτευχθεί υψηλό ποσοστό ακρίβειας του ακουστικού μοντέλου.

Το κύριο αρχείο παραμέτρων, που ελέγχει την όλη διαδικασία είναι το `sphinx_train.cfg` όπου ορίζονται και οι διάφορες παράμετροι εκπαίδευσης του μοντέλου μας. Το αρχείο [sphinx_train.cfg](#), που χρησιμοποιήθηκε κατά την διάρκεια των εργασιών, δίνεται στο παράρτημα της Μεταπτυχιακής διατριβής.

2.3.4 Τροφοδοσία sphinxtrain με υλικό εκπαίδευσης

Μετά την προετοιμασία όλων των απαραίτητων αρχείων ξεκινάμε το sphinxtrain, με την εντολή `sphinxtrain run`, το οποίο με την σειρά του ξεκινά τον αλγόριθμο εκπαίδευσης του μοντέλου μας. Στο τέλος της διαδικασίας πέρα από την δημιουργία των στατιστικών αρχείων που στο σύνολο τους αποτελούν το ακουστικό μοντέλο, έχουμε και την δημιουργία του αρχείου `align` που μας επιτρέπει να ελέγξουμε το πόσο ακριβές είναι το μοντέλο μας.

Μερικές ενδεικτικές καταγραφές από το αρχείο align δίνονται παρακάτω :

Αρχικό κείμενο 1

στα χρόνια του φιλίππου ΒΗΤΑ ΙΔΡΥΘΗΚΕ στην αθήνα το σωματείο γελωτοποιών ΠΟΥ ΕΙΧΕ την ονομασία *** ΕΞΗΚΟΝΤΑ από τον αριθμό των ΜΕΛΩΝ του πρόκειται για το πρώτο ΣΩΜΑΤΕΙΩΝ ηθοποιών της ιστορίας (129_Keti-Rovirou_I-ellinikotita-tis-Makedonias-000_000015_001380)

Κείμενο που αναγνωρίστηκε 1

στα χρόνια του φιλίππου ΔΙΚΙΑ ΓΕΝΝΗΘΗΚΕ στην αθήνα το σωματείο γελωτοποιών *** ΓΙΑ την ονομασία ΞΕΙ ΕΙΚΟΝΕΣ από τον αριθμό των ΜΕΡΩΝ του πρόκειται για το πρώτο ΣΩΜΑΤΕΙΟ ηθοποιών της ιστορίας (129_Keti-Rovirou_I-ellinikotita-tis-Makedonias-000_000015_001380)

Αποτελέσματα 1

Words: 30 Correct: 23 Errors: 8 Percent correct = 76.67% Error = 26.67%
Accuracy = 73.33% Insertions: 1 Deletions: 1 Substitutions: 6

Αρχικό κείμενο 2

ομιλητή ΣΕ ΉΝΑ ΑΚΡΟΑΤΗΡΙΟ ΚΑΙ αυτό που ΖΟΥΜΕ αυτή τη στιγμή ΤΙ θα το *** *** ΛΕΓΑΜΕ ΔΗΜΟΣΙΑ Η ΙΔΙΩΤΙΚΗ σφαίρα (67_Nikoleta-Tsitsanudi_O-sighronos-dimosiografikos-logos-ke-ta-xarakteristika-tu-138_160416_161121)

Κείμενο που αναγνωρίστηκε 2

ομιλητή ΕΙΣΑΙ ΑΝΕΤΑ ΝΑ ΠΡΟΤΕΙΝΩ αυτό που ΖΟΥΝ αυτή τη στιγμή ΔΕΝ θα το ΛΕΓΑ ΜΕ ΤΗ ΓΝΩΣΗ ΕΙΤΕ ΤΗ σφαίρα (67_Nikoleta-Tsitsanudi_O-sighronos-dimosiografikos-logos-ke-ta-xarakteristika-tu-138_160416_161121)

Αποτελέσματα 2

Words: 19 Correct: 9 Errors: 12 Percent correct = 47.37% Error = 63.16% Accuracy = 36.84% Insertions: 2 Deletions: 0 Substitutions: 10

Η συνολική ακρίβεια του μοντέλου μας προκύπτει από τον συνδυασμό όλων των επιμέρους στατιστικών. Έχοντας τα επιμέρους στατιστικά, μπορούμε να επέμβουμε στις παραμέτρους εκπαίδευσης και το υλικό και στην συνέχεια να έχουμε ξανά διαδικασίες εκπαίδευσης μέχρι να πετύχουμε καλύτερα ποσοστά ακρίβειας (<https://cmusphinx.github.io/wiki/tutorialtuning/>). Είναι κατανοητό επίσης ότι έχουμε την ευχέρεια να εκπαιδεύσουμε μια σειρά από μοντέλα αναλόγως την εφαρμογή που θέλουμε να αναπτύξουμε. Ο γενικός κανόνας που ισχύει είναι ότι περιορισμένα λεξιλόγια είναι πιο εύκολο και γρήγορο να εκπαιδευτούν, παρουσιάζοντας την ίδια στιγμή καλές ακρίβειες.

Η Μεταπτυχιακή αυτή διατριβή, δημιούργησε τόσο τον πυρήνα του Ελληνικού μοντέλου, για την πλατφόρμα του CMU Sphinx, όσο και ένα πιο εξειδικευμένο για χρήση σε εργασίες ROV ([Remotely Operated Vechicles](#)) μοντέλο.

Τα δυο Ελληνικά μοντέλα βρίσκονται ελεύθερα διαθέσιμα στις παρακάτω διαδικτυακές διευθύνσεις (αποθετήρια Στρατιωτικής Σχολής Ευελπίδων και CMU Sphinx) :

<https://gitlab.sse.gr/fpantazoglou/omilia>

<https://goo.gl/9v3QqG> (Εικόνα 26),

Το ποιο εξειδικευμένο και μικρό μοντέλο που δημιουργήθηκε, παρουσιάζεται πιο αναλυτικά στο Κεφάλαιο 4.

Home / Browse / CMU Sphinx / Files

CMU Sphinx

Speech Recognition Toolkit
Brought to you by: air, arthchan2003, awb, bhiksha, and 5 others

Summary | **Files** | Reviews | Support | Forums | Code | Issues | Mailing Lists

Looking for the latest version? [Download sphinx4-5prealpha-src.zip \(41.3 MB\)](#)

Home / Acoustic and Language Models / Greek

Name	Modified	Size	Downloads / Week
Parent folder			
cmusphinx-el-gr-5.2.tar.gz	2017-05-12	102.4 MB	25
README.txt	2017-05-12	201 Bytes	4
Totals: 2 Items		102.4 MB	29

This directory contains generic Greek acoustic model trained with latest sphinxtrain, language model and a dictionary.

This model is created with the help and due to Fotis Pantazoglou <fotis@hcmr.gr>

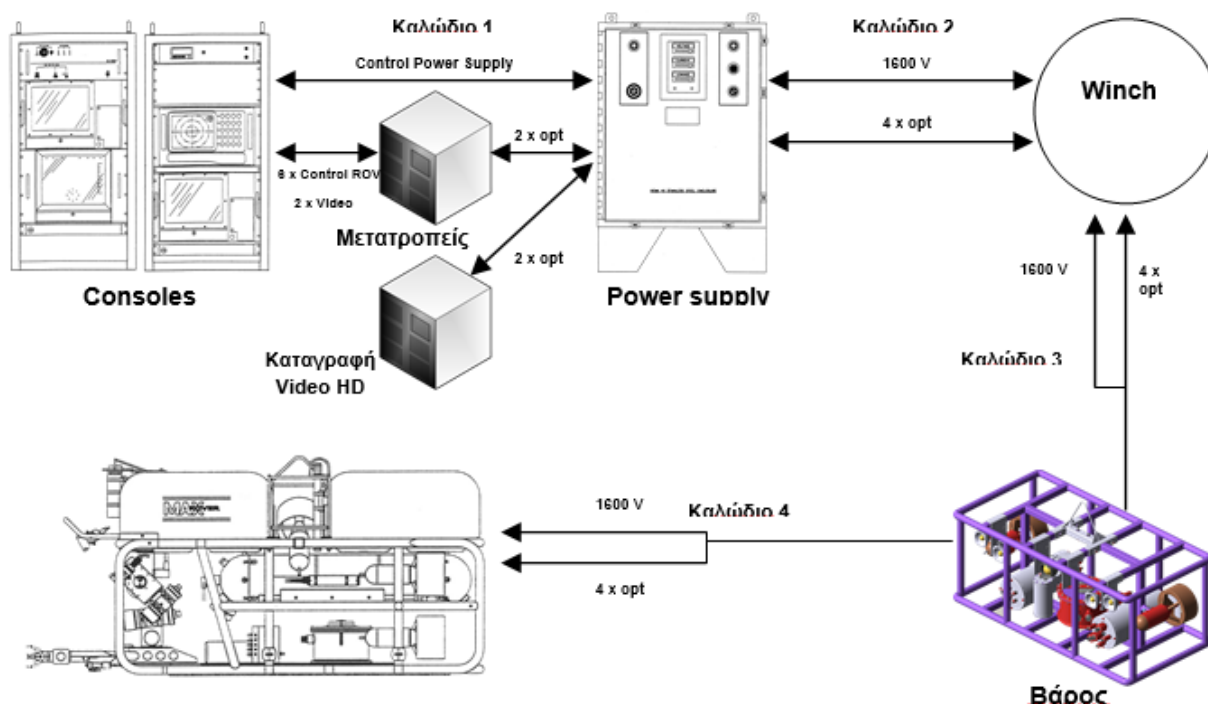
Εικόνα 26. Το Ελληνικό μοντέλο στο sourceforge.

ΚΕΦΑΛΑΙΟ 4

Σενάρια χρήσης

§1. Χειρισμός βραχιόνων σε αυτόνομα υποβρύχια οχήματα (ROV)

Τα αυτόνομα υποβρύχια οχήματα (Remotely Operated Vehicles) είναι μια οικογένεια υποβρυχίων ρομποτικών οχημάτων τα οποία μας επιτρέπουν να εργαζόμαστε σε συνθήκες και περιβάλλοντα που δεν είναι είτε φιλικά, είτε εύκολα προσιτά από τον άνθρωπο (<http://www.rov.org/>). Η τυπική διαμόρφωση ενός ολοκληρωμένου συστήματος ROV περιλαμβάνει τα παρακάτω μέρη και δίνεται στην Εικόνα 27:



Εικόνα 27. Διαμόρφωση συστήματος Max Rover ROV
(Λεωνίδας Μανουσάκης 2012)

- Το αυτόνομο υποβρύχιο ρομποτικό όχημα.
- Το σύστημα διασύνδεσης εύκαμπτου καλωδίου με το άκαμπτο καλώδιο. Σε πολλές περιπτώσεις αυτό το σύστημα είναι επίσης ένα αυτόνομο υποβρύχιο ρομποτικό όχημα, ενώ σε άλλες είναι απλά μια κατασκευή με βάρους και κουτιά διασύνδεσης καλωδίων.
- Το βίντσι που είναι υπεύθυνο για την κατάδυση και ανάδυση του άκαμπτου καλωδίου αλλά και του υποβρυχίου ρομποτικού οχήματος.
- Τα καλώδια διασύνδεσης, που λειτουργούν ουσιαστικά σαν ομφάλιος λώρος ανάμεσα στο ROV και το πλοίο επιφανείας. Μεταφέρουν την ισχύ την επικοινωνία και τα δεδομένα από και προς το ROV (Εικόνα 28).
- Το σύστημα μετασχηματισμού και παροχής της τάσης του ρεύματος.
- Τις κονσόλες ελέγχου χειρισμού και καταγραφής του συστήματος (Εικόνα 29).
- Το πλοίο επιφανείας που υποστηρίζει τις εργασίες του ROV. Εναλλακτικά φυσικά αυτές μπορούν να διενεργηθούν και από σταθερές πλατφόρμες εργασίας (πχ. Πλατφόρμες εξόρυξης πετρελαίου).



Εικόνα 28. Τύποι καλωδίων που χρησιμοποιούνται στα ROV
(<http://www.rov.org>)



Εικόνα 29. Κονσόλες χειρισμού Max Rover ROV

Ο διαχωρισμός των ROV σύμφωνα με την επιτροπή αυτόνομων υποβρυχίων οχημάτων δίνεται στον πίνακα που ακολουθεί.

<u>Class</u>	<u>Type</u>	<u>Power (hp)</u>
Micro Observation (<100 meters)	Low Cost Small Electric ROV	<5
Mini Observation (< 300 meters)	Mini (Small & #9;(Electric))	<10
Light/Medium Work Class (<2,000 meters)	Medium (Electro/Hyd)	<100
Observation/Light Work Class (< 3,000 meters)	High Capacity Electric	<20
Heavy Work Class /Large Payload (<3,000 meters)	High Capacity (Electro/Hyd)	<300
Observation/Data Collection (>3,000 meters)	Ultra-Deep (Electric)	<25
Heavy Work Class /Large Payload (>3,000 meters)	Ultra-Deep (Electro/Hyd)	<120
Trenching and Burial	Bottom Crawlers and Plows	
Towed Systems	Towed Systems	
Autonomous Underwater Vehicles	Untethered AUVs	

Είναι εύκολα κατανοητό ότι ύπαρξη πολλαπλών τεχνολογικών λύσεων έχει ωθήσει τον σχεδιασμό σε νέους δρόμους. Έτσι καθημερινά έχουμε εξελίξεις στο θέμα σχεδιασμού αισθητήρων, βραχιόνων, εργαλείων, συστημάτων ελέγχου, συστημάτων χειρισμού συστημάτων τροφοδοσίας, προπελών, συστημάτων οπτικής καταγραφής συστημάτων sonar κτλ.

Ένα από τα πιο πολύτιμα εργαλεία ενός ROV, είναι το σύστημα του η των βραχιόνων που αυτό φέρει. Ουσιαστικά είναι το σύστημα που μας επιτρέπει να εργαστούμε με την βοήθεια του ROV σε περιβάλλοντα που δεν είναι προσβάσιμα από εμάς. Τα συστήματα βραχιόνων είναι τηλεχειριζόμενα από τους χειριστές που βρίσκονται στο σκάφος επιφανείας και προφανώς δεν είναι το ίδιο με τους βιομηχανικούς βραχίονες οι οποίοι απλά εκτελούν μια προγραμματισμένη κίνηση συνεχώς. Αποτελούν ουσιαστικά την «προέκταση» των χεριών του χειριστή και είναι σε θέση σε πραγματικό χρόνο να εκτελέσουν εργασίες με διαφορετικούς βαθμούς δυσκολίας και πολυπλοκότητας. Το πιο βασικό σύστημα βραχίονα αποτελείται από μια συσκευή χειρισμού και έναν βραχίονα στην άκρη του οποίου μπορούμε να προσαρμόσουμε διαφορετικά εργαλεία (Εικόνα 30). Στα πιο εξελιγμένα συστήματα υπάρχουν πολύπλοκα συστήματα ηλεκτρονικού ελέγχου.



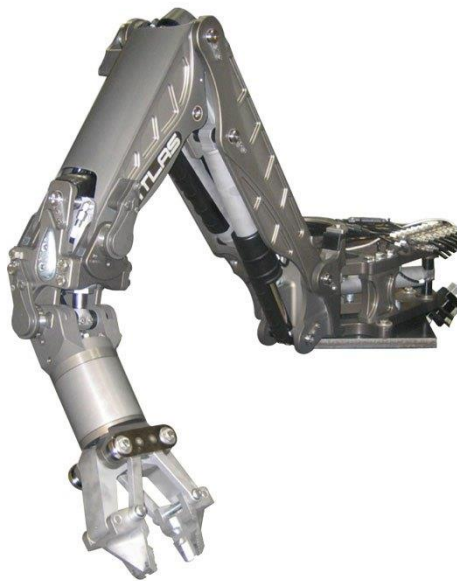
Εικόνα 30. Εργαλεία για βραχίονες Rov
(<http://www.rov.org>)

Δεδομένων των αυξημένων απαιτήσεων αλλά και των τεχνολογικών αλμάτων που διαπιστώνονται στον χώρο, αυτή την στιγμή υπάρχουν μια σειρά από διαφορετικά συστήματα βραχιόνων στην αγορά των ROV. Ιδίως σε τομείς όπως η αγορά πετρελαίων όπου οι απαιτήσεις εργασιών από ROV είναι αυξημένες, έχουμε την χρήση βραχιόνων με πολλαπλούς βαθμούς ελευθερίας που μας επιτρέπουν την διενέργεια πολύ εξειδικευμένων τεχνικών εργασιών.

Ένα από τα πιο σημαντικά μέρη του συστήματος βραχιόνων είναι το σύστημα χειρισμού που ο χειριστής χρησιμοποιεί ώστε να πραγματοποιήσει τις εργασίες που επιθυμεί. Τα συστήματα χειρισμού διαφέρουν ανάλογα με την εταιρεία και ανάλογα με τον βαθμό ελευθερίας του βραχίονα που έχουν να διαχειριστούν. Έτσι για παράδειγμα έχουμε συστήματα όπου ο χειρισμός γίνεται με την χρήση απλών λεβιέδων (Εικόνα 31), όπως το σύστημα της εταιρείας Hydrolek (<http://www.hydro-lek.com/>), η πιο εξελιγμένα συστήματα χειρισμού σαν και αυτό που χρησιμοποιεί η εταιρεία ρομποτικών βραχιόνων Schilling Robotics (www.fmctechnologies.com) (Εικόνα 32).



Εικόνα 31. Χειριστήρια συστήματος βραχιόνων Hydrolek



Εικόνα 32. Χειριστήρια συστήματος βραχιόνων Schilling
(<http://www.fmctechnologies.com>)

Αυτό που γίνεται εύκολα κατανοητό είναι, ότι ο χειρισμός τέτοιων συστημάτων απαιτεί εμπειρία επιδεξιότητα και καλή γνώση του συστήματος από την μεριά του χειριστή. Συνήθως σε εργασίες ROV η συνήθης βάρδια εργασίας είναι 6 ώρες κάτι που προσθέτει περαιτέρω βαθμό δυσκολίας στην όλη διαδικασία. Οι κινήσεις από την μεριά του χειριστή θα πρέπει να χαρακτηρίζονται από ακρίβεια ενώ ταυτόχρονα θα πρέπει να μπορεί να επαναλάβει ορισμένες

από αυτές με το ίδιο τρόπο. Και όλα αυτά εργαζόμενος μέσα σε ένα περιβάλλον εργασίας το οποίο δεν είναι σταθερό, καθώς αυτό μεταβάλλεται διαρκώς (Εικόνα 33).

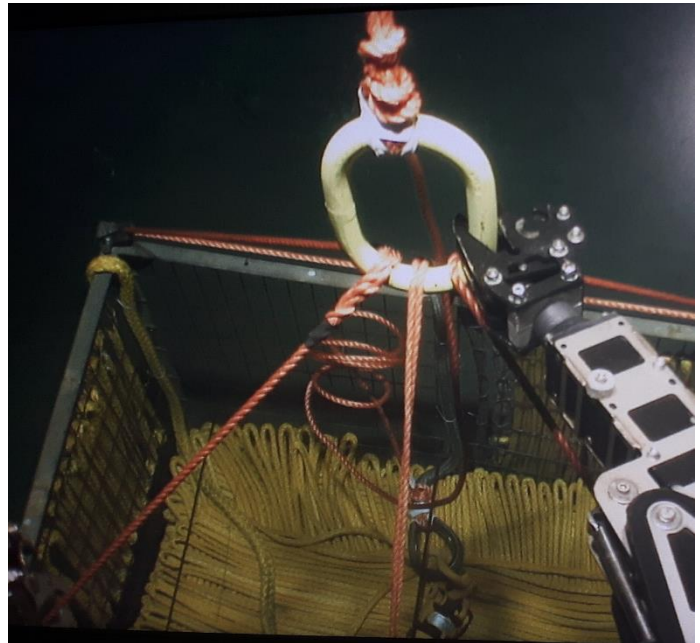
Στο πλαίσιο αυτό ένα σύστημα το οποίο θα συνδυάζει την απλότητα στον έλεγχο, την ευκολία χρήσης, την επανάληψη θα μπορούσε να συνδράμει αποφασιστικά στην περαιτέρω αύξηση της παραγωγικότητας των εργαλείων που χρησιμοποιούνται από τους βραχίονες των ROV. Σαν μια πρώτη αλλαγή στα υπάρχοντα συστήματα ελέγχου και χειρισμού θα ήταν η χρήση φωνητικών εντολών αντί των χειριών του χειριστή.

Οι φωνητικές εντολές είναι σαφώς πιο απλές και μπορούν υπό προϋποθέσεις να βελτιώσουν σε δραματικό βαθμό την ευκολία χειρισμού των βραχιόνων.

Έτσι στα πλαίσια των εργασιών και δοκιμών αυτής της Μεταπτυχιακής διατριβής, και μετά την ολοκλήρωση του Ελληνικού μοντέλου για το CMU Sphinx, δημιουργήθηκε ένα τροποποιημένο και πιο μικρό μοντέλο, έχοντας κατά νου ότι αυτό θα χρησιμοποιούταν σε εργασίες ROV.

Για τον λόγο αυτό, έγινε χρήση του ήδη υπάρχοντος υλικού το οποίο προερχόταν από την ηχογράφηση 30 φράσεων, από φυσικά πρόσωπα, οι οποίες σε γενικές γραμμές συμπεριελάμβαναν τις πιο πολλές από τις δυνητικές εντολές που θα χρειαζόταν ένας χειριστής συστήματος ROV. Έγινε προσπάθεια να υπάρχει σχεδόν ίσως αριθμός ανδρών και γυναικών κατά τις ηχογραφήσεις, αν και συνήθως σαν χειριστές συστημάτων ROV συναντάμε κυρίως άνδρες.

Στην συνέχεια και πάντα σύμφωνα με την ήδη ισχύουσα πρακτική «εκπαίδευσης» του CMU Sphinx οι ήχοι αφού επεξεργάστηκαν, τροφοδότησαν το σύστημα με αποτέλεσμα να δημιουργηθεί ένα πιο μικρό Ελληνικό μοντέλο προσαρμοσμένο στις ανάγκες που προκύπτουν κατά τις εργασίες ROV. Τα αποτελέσματα της διαδικασίας εκπαίδευσης του μικρότερου και σχετικά ορισμένου Ελληνικού μοντέλου φαίνονται εν συντομία στην Εικόνα 34 ενώ παρατίθενται αναλυτικά στο [Παράρτημα](#) της Μεταπτυχιακής διατριβής.



Εικόνα 33. Εργασίες βραχίονα Hydrolek σε βάθος 1000 m


```

στρίψε δεξιά αριστερά σιγά σιγά (user-488)
στρίψε δεξιά αριστερά σιγά σιγά (user-488)
Words: 5 Correct: 5 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%
Insertions: 0 Deletions: 0 Substitutions: 0
σταμάτα ξεκίνα συνέχισε κάνε πίσω (user-489)
σταμάτα ξεκίνα συνέχισε κάνε πίσω (user-489)
Words: 5 Correct: 5 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%
Insertions: 0 Deletions: 0 Substitutions: 0
αύξησε ταχύτητα μείωσε ταχύτητα (user-490)
αύξησε ταχύτητα μείωσε ταχύτητα (user-490)
Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%
Insertions: 0 Deletions: 0 Substitutions: 0
εκατόν ογδόντα μοίρες (user-491)
εκατόν ογδόντα μοίρες (user-491)

```

Εικόνα 34. Αποτελέσματα εκπαίδευσης Ελληνικού μοντέλου προσαρμοσμένο για χρήση σε εργασίες ROV

Είναι κατανοητό ότι η ακρίβεια κατά την αναγνώριση κινείται σε απόλυτα σχεδόν επίπεδα. Θα πρέπει βέβαια να τονιστεί ότι όπως και στο γενικό μοντέλο έτσι και σε αυτό η ακρίβεια αναγνώρισης επηρεάζεται από παράγοντες όπως ο θόρυβος του περιβάλλοντος χώρου, η ορθή εκφορά του λόγου από τον ομιλητή, και το στρες του ομιλητή κατά την εκφορά του λόγου.

Καθώς ήδη η βιομηχανία σχεδιασμού και ανάπτυξης συστημάτων βραχιόνων για ROV κινείται προς την εξεύρεση πιο αυτοματοποιημένων λύσεων, το Ελληνικό μοντέλο αποτελεί ήδη μια έτοιμη πλατφόρμα πάνω στην οποία μπορεί να δομηθεί ένα νέο σύστημα ελέγχου. Δίνοντας την απαραίτητη ευελιξία και χρηστικότητα μπορεί υπό τις κατάλληλες προϋποθέσεις να οδηγήσει τα συστήματα αυτά σε νέες πιο παραγωγικές κατευθύνσεις, που ενδεχομένως να είναι και πιο οικονομικές από τις ήδη υπάρχουσες.

Το συγκεκριμένο Ελληνικό μοντέλο βρίσκεται ελεύθερα διαθέσιμο, αποκλειστικά στο αποθετήριο της Στρατιωτικής Σχολής Ευελπίδων (<https://gitlab.sse.gr/fpantazoglou/omilia>).

§2. Εφαρμογές σε περιβάλλοντα στρατιωτικού χαρακτήρα

Η φωνητική αναγνώριση και εντολές στην σημερινή εποχή έχουν ήδη εφαρμοστεί σε μια σειρά περιβαλλόντων δράσης, στρατιωτικού χαρακτήρα (Skaff, 2010). Τα πλεονεκτήματα που αυτή προσφέρει έχουν ήδη αξιολογηθεί από οργανισμούς στρατιωτικού χαρακτήρα όπως το ΝΑΤΟ με αποτέλεσμα ήδη από την δεκαετία του 1970 να έχουν δημιουργηθεί ομάδες έρευνας και εργασίας (Pigeon, 2005) που μέχρι και σήμερα εργάζονται στο τομέα εφαρμογής της τεχνολογίας φωνητικής αναγνώρισης σε περιβάλλοντα στρατιωτικού χαρακτήρα. Δυστυχώς σε αυτή την προσπάθεια δεν υπάρχει Ελληνική συμμετοχή, οπότε το δημιουργημένο Ελληνικό μοντέλο θα μπορούσε να συμπεριληφθεί σε μελλοντικές εφαρμογές με Ελληνική συμμετοχή και ενδιαφέρον.

Προφανώς και οι εφαρμογές που δρομολογούνται σε στρατιωτικά περιβάλλοντα έχουν να αντιμετωπίσουν πολύ διαφορετικές προκλήσεις από τις αντίστοιχες εφαρμογές που χρησιμοποιούμε στην πολιτική μας ζωή. Για παράδειγμα, η αυτόματη φωνητική αναγνώριση που χρησιμοποιείται σε στρατιωτικά σενάρια πρέπει να είναι ανθεκτική στις δυσμενείς συνθήκες. Αυτό γιατί πολλές καταστάσεις στο στρατιωτικό περιβάλλον έχουν σαν



Εικόνα 35. Cockpit ελικοπτέρου NH-90 (<http://www.nhindustries.com>)

χαρακτηριστικό την ύπαρξη δύσκολων ακουστικών συνθηκών όπου η ύπαρξη υψηλού θορύβου, κραδασμών ή υψηλών δυνάμεων g είναι συνηθισμένη. Σύμφωνα με τον (Pigeon, 2005) οι στρατιωτικές εφαρμογές που κάνουν χρήση της συγκεκριμένης τεχνολογίας μπορούν να ενταχθούν στις παρακάτω κατηγορίες:

- **Διοίκηση και έλεγχος**

Τα συστήματα ελέγχου και διοίκησης μπορούν να βοηθηθούν από την ανθρώπινη αλληλεπίδραση με τους υπολογιστές, τα όπλα και τα συστήματα αισθητήρων με φωνή. Αυτό βέβαια προϋποθέτει πολύ υψηλά επίπεδα απόδοσης της τεχνολογίας φωνής και γλώσσας, υπό δυσμενείς συνθήκες, σε πραγματικό χρόνο, όπως κίνηση και θόρυβος και διάφορα φαινόμενα στρες λειτουργώντας με πολύγλωσση είσοδο και έξοδο. Στο περιβάλλον εργασίας για παράδειγμα ενός σύγχρονου ελικοπτήρου όπως το NH-90 (Εικόνα 35) ο χειριστής έχει σημαντικό και απαιτητικό φόρτο εργασίας κατά την διάρκεια της αποστολής του. Μέσα σε ένα περιβάλλον που χαρακτηρίζεται από υψηλό θόρυβο και κραδασμούς είναι επιφορτισμένος τόσο με την πτητική διαδικασία αυτή κάθε αυτή όσο και με την παρακολούθηση όλων των συστημάτων αισθητήρων που το ελικόπτερο διαθέτει. Μέσα σε αυτό το πλαίσιο λειτουργίας εταιρείες όπως η Thales Ανιονics ήδη πειραματίζονται από την δεκαετία του 1990 σε συστήματα διαχείρισης τα οποία είναι βασισμένα στην δια δράση μεταξύ πιλότου και μηχανής. Έτσι ο πιλότος θα μπορεί να λαμβάνει πληροφορίες για την κατάσταση του πτητικού μέσου μιλώντας στην μηχανή, αφαιρώντας σημαντικό φόρτο εργασίας από πάνω του και επιτρέποντας του να επικεντρωθεί σε εργασίες που αυτός εκείνη την στιγμή επιθυμεί. (πχ. παρακολούθηση της υπέρυθρης κάμερας παρακολούθησης του αεροσκάφους).

- **Επικοινωνίες**

Οι επικοινωνίες πρέπει να λειτουργούν με ασφάλεια, με υψηλή ευκρίνεια, υπό συνθήκες θορύβου και εμπλοκής. Το σήμα ομιλίας, για παράδειγμα, πρέπει να κωδικοποιηθεί και να μεταδοθεί με αρκετή πίστη ώστε να γίνει κατανοητό από ακροατές που δεν είναι γνώστες της γλώσσας που μιλιέται.

- **Ηλεκτρονικοί υπολογιστές και πρόσβαση σε πληροφορίες**

Την σημερινή εποχή οι υπολογιστές και η πρόσβαση στις πληροφορίες είναι ένα κρίσιμο κομμάτι των σύγχρονων στρατιωτικών επιχειρήσεων. Η τεχνολογία φωνητικής αναγνώρισης μπορεί να χρησιμοποιηθεί από το στρατιωτικό προσωπικό ώστε αυτό να ελέγξει διερευνήσει υπολογιστές και πληροφορίες κάνοντας χρήση της φωνής. Μια πολύ χρήσιμη δυνατότητα ιδίως για προσωπικό που έχει απασχολημένα τα χέρια και μάτια

του την στιγμή που προσπαθεί να ανταπεξέλθει σε μεγάλο φόρτο εργασίας. Ταυτόχρονα η όλο και εξελισσόμενη νέες συμμαχίες θέτουν νέες απαιτήσεις σε θέματα πολύ επίπεδης ασφάλειας συστημάτων πληροφοριών καθώς η πρόσβαση στις διάφορες πληροφορίες πρέπει να περιοριστεί στα κατάλληλα επίπεδα. Έτσι οι απαιτήσεις σχετικά με την τεχνολογία φωνητικής αναγνώρισης, περιλαμβάνουν επαλήθευση ομιλητών, πιστοποίηση δεδομένων ήχου, πολύγλωσση εισαγωγή δεδομένων και δυνατότητα μετάφρασης ή σύνταξης των πληροφοριών από τη μία γλώσσα στην άλλη.

- **Συλλογή πληροφοριών**

Η συλλογή πληροφοριών θέτει υψηλές απαιτήσεις στην επεξεργασία και τη διάδοση των πληροφοριών. Προκειμένου οι πληροφορίες να είναι χρήσιμες, πρέπει να είναι υψηλής αξίας, ακριβείς και καταγεγραμμένες έγκαιρα. Η έκρηξη στην ανάπτυξη της τεχνολογίας των επικοινωνιών και του διαδικτύου μας έχει προσφέρει μια τεράστια πηγή με δεδομένα ήχου και κειμένου. Τα δεδομένα αυτά θα πρέπει να φιλτραριστούν γιατί ενδεχομένως να περιέχουν και πληροφορίες με υψηλή στρατηγική ή στρατιωτική αξία. Η όλη διαδικασία λόγω του τεράστιου όγκου δεδομένων που υπάρχουν μπορεί να αποσυμφωρηθεί με την χρήση εφαρμογών που εμπεριέχουν την τεχνολογία της αυτόματης φωνητικής αναγνώρισης.

- **Εκπαίδευση δυνάμεων**

Η εκπαίδευση δυνάμεων για στρατιωτικές επιχειρήσεις μπορεί να βοηθηθεί σημαντικά εφαρμόζοντας την τεχνολογία ομιλίας ώστε να επιτρέψει στους ανθρώπους να αλληλοεπιδρούν με προηγμένα συστήματα προσομοίωσης κάνοντας χρήση φωνητικών εντολών. Επιπλέον, για τις πολυεθνικές επιχειρήσεις, είναι απαραίτητη η εκπαίδευση σε ξένες γλώσσες. Η εκπαίδευση αυτή μπορεί να υποστηριχθεί με τη χρήση τεχνολογιών ομιλίας και γλώσσας για την παροχή μάθησης ξένων γλωσσών με τη βοήθεια υπολογιστή προς το στρατιωτικό προσωπικό που συμμετάσχει στις πολυεθνικές επιχειρήσεις.

- **Πολυεθνικές δυνάμεις**

Οι επιχειρήσεις πολυεθνικών δυνάμεων απαιτούν το συντονισμό δυνάμεων που μιλούν διαφορετικές γλώσσες (Εικόνα 36). Εδώ, η κατανόηση του λόγου και της γλώσσας καθώς και οι τεχνολογίες μετάφρασης μπορούν να βελτιώσουν την αποδοτικότητα και την επιτυχία των επιχειρήσεων. Βέβαια οι απαιτήσεις που τίθενται είναι υψηλές. Έτσι οι εφαρμογές θα πρέπει να επικεντρωθούν σε συγκεκριμένους τομείς για μετάφραση και

ανταλλαγή πληροφοριών μεταξύ πολυεθνικών δυνάμεων υιοθετώντας συγκεκριμένη φρασεολογία και ορολογία.



Εικόνα 36. Συμμετοχή Ελληνικών δυνάμεων στην πολυεθνική άσκηση Αργοναύτης 2017

Γίνεται εμφανές ότι το όλο και εξελισσόμενο στρατιωτικό περιβάλλον απαιτεί πλέον να έχουμε στην διάθεση μας νέα εργαλεία ώστε να αυξήσουμε την αποδοτικότητα μας κατά την διενέργεια των επιχειρήσεων. Το Ελληνικού μοντέλο μπορεί να ενταχθεί σε οποιαδήποτε από τις παραπάνω προσπάθειες για ανάπτυξη εφαρμογής, βοηθώντας σημαντικά την προσπάθεια των Ελληνικών Ενόπλων Δυνάμεων για συνεχή βελτίωση των επιχειρησιακών τους δυνατοτήτων.

ΚΕΦΑΛΑΙΟ 5

Συμπεράσματα

Οι τεχνολογίες αναγνώρισης φωνής και χρήσης φωνητικών εντολών έχουν συμβάλει σημαντικά σε μια σειρά καθημερινών μας δραστηριοτήτων. Εκμεταλλευόμενοι την ραγδαία πλέον ανάπτυξη αλλά και διάδοση της τεχνολογίας έχουμε στην διάθεση μας μια σειρά εργαλείων που μας επιτρέπουν να καλυτερεύουμε τόσο την καθημερινότητά μας, όσο και την επίδοσή μας σε μια σειρά πολύ απαιτητικών εργασιών.

Η Μεταπτυχιακή διατριβή που εκπονήθηκε είχε σαν αρχικό σκοπό να συνδράμει στις προσπάθειες ανάπτυξης της ανοικτής πλατφόρμας εργαλείων CMU Sphinx. Έχοντας εντοπίσει την έλλειψη υποστήριξης που υπάρχει για την Ελληνική γλώσσα, γεγονός που καθιστούσε την χρήση του CMU Sphinx στην ανάπτυξη Ελληνικών εφαρμογών μη εφικτή, ο συγγραφέας εργάστηκε στο πλαίσιο που η πλατφόρμα έχει καθορίσει και ανέπτυξε μια σειρά εργαλείων τα οποία συνθέτουν στο σύνολό τους το ολοκληρωμένο γενικό Ελληνικό μοντέλο της πλατφόρμας.



Σχεδιάγραμμα 10. Τα μέρη του Ελληνικού μοντέλου για το CMU Sphinx

Η Ελληνική γλώσσα έχει τις ιδιαιτερότητες της και αυτό έθεσε έναν επιπλέον παράγοντα δυσκολίας στην όλη διαδικασία. Όπως περιεγράφηκε αναλυτικά στην παρούσα διατριβή, οι εργασίες και ο κώδικας που έπρεπε να αναπτυχθεί για την επιτυχή ολοκλήρωση του εγχειρήματος έπρεπε να ξεινηθούν από το σημείο μηδέν, καθώς δεν ήταν δυνατόν για την Ελληνική γλώσσα να γίνει χρήση ήδη υπάρχοντος κώδικα.

Πέρα από τις γενικές οδηγίες που αναφέρονται στο αποθετήριο του CMU Sphinx σημαντική πηγή έμπνευσης ως προς την προσέγγιση ανάπτυξης του ολοκληρωμένου γενικού Ελληνικού μοντέλου αποτέλεσε η δουλειά που έχει γίνει από τους (Tsardoulas et al., 2015). Οι διαδικασίες ανάπτυξης του Ελληνικού μοντέλου, το οποίο περιλαμβάνει το Ακουστικό μοντέλο, το Φωνητικό λεξικό και το Γλωσσικό μοντέλο (Σχεδιάγραμμα 10), ολοκληρώθηκαν χωρίς ιδιαίτερα προβλήματα, παραδίδοντας προς την διεθνή κοινότητα το γενικό Ελληνικό μοντέλο του CMU Sphinx.

Το γενικό Ελληνικό μοντέλο στην έκδοση που έχει ήδη αναρτηθεί στα αποθετήρια της Στρατιωτικής Σχολής Ευελπίδων (<https://gitlab.sse.gr/fpantazoglou/omilia>) αλλά και της ίδιας της πλατφόρμας (Εικόνα 36) (<https://goo.gl/9v3QqG>), περιλαμβάνει τις πιο γενικές και μεγάλες εκδόσεις του Γλωσσικού μοντέλου και του Φωνητικού λεξικού.

Φυσικά με βάση το υλικό που έχει ήδη δημιουργηθεί και αναρτηθεί είναι εύκολο να αναπτυχθούν και άλλα Ελληνικά μοντέλα τα οποία θα είναι προσαρμοσμένα σε συγκεκριμένο πλαίσιο το οποίο ορίζεται από τις ανάγκες μας. Έτσι για τις ανάγκες που προκύπτουν από τις εργασίες χειρισμού ενός συστήματος ROV έχει ήδη δημιουργηθεί ένα Ελληνικό μοντέλο ειδικά προσαρμοσμένο σε συγκεκριμένη φρασεολογία και ορολογία, υιοθετώντας εάν μικρότερο και πιο ακριβές Γλωσσικό μοντέλο. Ο ενδιαφερόμενος χρήστης μπορεί να το βρει ελεύθερα διαθέσιμο αποκλειστικά στο αποθετήριο της Στρατιωτικής Σχολής Ευελπίδων (<https://gitlab.sse.gr/fpantazoglou/omilia>).

Home / Acoustic and Language Models

Name *	Modified *	Size
↑ Parent folder		
■ Greek	2017-05-12	
■ Indian English	2017-04-26	
■ German	2017-01-08	
■ French	2017-01-04	
■ Dutch	2016-10-21	
■ Archive	2016-10-21	
■ US English	2016-10-05	
■ Spanish	2016-08-30	
■ Italian	2016-06-06	
■ Mandarin	2016-05-31	
■ Hindi	2016-05-22	
■ Kazakh	2016-05-22	
■ Russian	2016-05-07	

Εικόνα 37. Το σύνολο των διατιθέμενων μοντέλων της πλατφόρμας CMU Sphinx

Στον ίδιο χώρο έχουν αναρτηθεί και διάφορα αρχεία ανοικτού κώδικα που χρησιμοποιήθηκαν κατά την φάση των εργασιών της Μεταπτυχιακής διατριβής, αλλά και πληροφοριακό υλικό που ο συγγραφέας εκτιμά ότι μπορούν να βοηθήσουν τους ενδιαφερόμενους χρήστες του Ελληνικού μοντέλου.

Το επόμενο βήμα στις επιστημονικές αναζητήσεις του συγγραφέα θα αποτελέσει η προσπάθεια δημιουργίας μηχανισμού φωνητικού ελέγχου μηχανικών βραχιόνων, με την χρήση φωνητικών εντολών και μικρο επεξεργαστών της οικογένειας [Raspberry pi](#). Άλλωστε ήδη το Pocketsphinx υποστηρίζεται στο Raspberry pi και η απόδοση του κρίνεται άκρως ικανοποιητική τόσο για το γενικό και πιο μεγάλο Ελληνικό μοντέλο, όσο και για το μικρότερο και ειδικά προσαρμοσμένο σε εργασίες ROV μοντέλο.

Ολοκληρώνοντας την Μεταπτυχιακή αυτή διατριβή ο συγγραφέας θα ήθελε να καταθέσει την βεβαιότητα του, ότι η όλη προσπάθεια θα συνεισφέρει σημαντικά στην πλατφόρμα CMU Sphinx αλλά και στις προσπάθειες Ελλήνων προγραμματιστών για ανάπτυξη εφαρμογών που χρησιμοποιούν την συγκεκριμένη ανοιχτή πλατφόρμα. Η συγκεκριμένη εργασία έρχεται σε πρώτη φάση να καλύψει ένα κενό που είχε το CMU Sphinx αλλά και ταυτόχρονα να αποτελέσει εφαλτήριο μελλοντικών εργασιών που θα εκπορεύονται από αυτή την πρώτη προσπάθεια.

Η συνολική προσπάθεια προφανώς συνεχίζεται με τα παράγωγά (Ελληνικό μοντέλο και επιμέρους στοιχεία του) της, να μπορούν να βελτιωθούν, να παραμετροποιηθούν και να επεκταθούν σύμφωνα με τις εκάστοτε απαιτήσεις των εφαρμογών μας.

Άλλωστε οι προσπάθειες βελτίωσης των τρόπων δια δράσης ανάμεσα σε άνθρωπο και μηχανή, είναι καταδικασμένες πάντα να προχωράνε άλλοτε με μικρά βήματα άλλοτε με άλματα εναρμονισμένες συνήθως με τους αέναους κύκλους της ανθρώπινης ζωής.

Φώτιος Πανταζόγλου
Ηράκλειο-Οκτώβριος 2017

Βιβλιογραφία

Aegisub Advanced Subtitle Editor. (n.d.). Retrieved May 12, 2017, from <http://www.aegisub.org/>

Anusuya, M., & Katti, S. (2009). Speech recognition by machine: A review. *International Journal of Computer Science and Information Security*, 6(3), 181–205.
<https://doi.org/10.1109/PROC.1976.10158>

Audacity® | Free, open source, cross-platform audio software for multi-track recording and editing. (n.d.). Retrieved May 6, 2017, from <http://www.audacityteam.org/>

Berners-Lee, T., Cailliau, R., Luotonen, A., Frystyk Nielsen, H., & Secret, A. (1994). The World-Wide Web. *Communications of the ACM*.
<https://doi.org/10.1145/179606.179671>

Bisani, M., & Ney, H. (2008). Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5), 434–451.
<https://doi.org/10.1016/j.specom.2008.01.002>

Campbell-Kelly, M., Garcia-Swartz, D. D., & Campbell-Kelly, M. (2013). The history of the internet: the missing narratives. *Journal of Information Technology*, 28, 18–33.
<https://doi.org/10.1057/jit.2013.4>

CMU Sphinx. (n.d.). Retrieved May 11, 2017, from <http://cmusphinx.sourceforge.net/>

CMU Sphinx - Browse / Acoustic and Language Models at SourceForge.net. (n.d.). Retrieved May 11, 2017, from [https://sourceforge.net/projects/cmusphinx/files/Acoustic and Language Models/](https://sourceforge.net/projects/cmusphinx/files/Acoustic%20and%20Language%20Models/)

CMU Sphinx - Greek at SourceForge.net. (n.d.). Retrieved May 15, 2017, from

[https://sourceforge.net/projects/cmusphinx/files/Acoustic and Language Models/Greek/](https://sourceforge.net/projects/cmusphinx/files/Acoustic%20and%20Language%20Models/Greek/)

cmuclmtk at SourceForge.net. (n.d.). Retrieved May 11, 2017, from
<https://sourceforge.net/projects/cmusphinx/files/cmuclmtk/0.7/>

Controller, M. (n.d.). 1 System Overview, (Figure 1), 49–86.

Daines, D. H. (2011). An Architecture for Scalable , Universal Speech Recognition Thesis
Committee : c 2011 David Huggins Daines, 131. Retrieved from
http://www.lti.cs.cmu.edu/research/thesis/2011/david_hugginsdaines.pdf

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische
Mathematik*, 1(1), 269–271. <https://doi.org/10.1007/BF01386390>

Englund, C. (2004). Speech Recognition in the JAS 39 Gripen Aircraft - Adaptation to
Speech at Different G-loads, (March), 1–57.

Eurofighter Typhoon | The world's most advanced fighter aircraft. (n.d.). Retrieved May
3, 2017, from <https://www.eurofighter.com/the-aircraft#cockpit>

Gaikwad, S. K., Gawali, B. W., & Yannawar, P. (2010). A Review on Speech Recognition
Technique. *International Journal of Computer Applications*, 10(3), 16–24.
<https://doi.org/10.5120/1462-1976>

Hemdal, J. F., & G. W. Hughes. (1967). A feature based computer recognition program
for the modeling of vowel perception. *Models for the Perception of Speech and Visual Form*,
MIT Press(Models for the Perception of Speech and Visual Form).

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A., Jaitly, N., ... Kingsbury, B.
(2012). Deep Neural Networks for Acoustic Modeling in Speech Recognition. *Ieee
Signal Processing Magazine*, (November), 82–97.

<https://doi.org/10.1109/MSP.2012.2205597>

Home - RAPP project. (n.d.). Retrieved May 11, 2017, from <http://rapp-project.eu/>

Huggins-Daines, D., Kumar, M., Chan, A., Black, A. W., Ravishankar, M., & Rudnick, A. I. (2006). Pocketsphinx: A Free, Real-Time Continuous Speech Recognition System for Hand-Held Devices. *Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing*, 1, 185–188.

<https://doi.org/10.1109/ICASSP.2006.1659988>

Installing CMU-Sphinx on Ubuntu. (n.d.). Retrieved May 11, 2017, from <http://jrmeyer.github.io/installation/2016/01/09/Installing-CMU-Sphinx-on-Ubuntu.html>

iOS - Siri - Apple. (n.d.). Retrieved May 3, 2017, from <https://www.apple.com/ios/siri/>

jEdit - Programmer's Text Editor. (n.d.). Retrieved May 12, 2017, from <http://jedit.org/>

Karyotis, K.-V. (2016). *The hidden Markov model in speech recognition*.

Lamere, P., Kwok, P., Gouvea, E., Raj, B., Singh, R., Walker, W., ... Wolf, P. (2003). The CMU SPHINX-4 speech recognition system. *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2003)*, Hong Kong, 1, 2–5.

Leiner, B. M., Cerf, V. G., Clark, D. D., Kahn, R. E., Kleinrock, L., Lynch, D. C., ... Wolff, S. S. (1997). The past and future history of the internet. *COMMUNICATIONS OF THE ACM*, 40(2). Retrieved from <http://bnrg.eecs.berkeley.edu/~randy/Courses/CS294.S13/1.1x.pdf>

Marangozis, J. (2007). *An introduction to Minoan linear A*. LINCOM Europa.

Martin, T. B., Zadell, H. J., Grunza, E. F., Hmcher, M. B., & Reddy, D. R. (1974). Speech Recognition by Machine: A Review. *PROCEEDINGS OF THE IEEE Conf. Rec. Soc.*

Amcr. ZYans. Computers Tech. Rep, 6443(2), 541–546.

Odlyzko, A. (2000). The history of communications and its implications for the Internet.

Social Science Research Network, 6, 1–28. <https://doi.org/10.2139/ssrn.235284>

Open Courses. (n.d.). Retrieved May 3, 2017, from <http://opencourses.gr/>

Park, K.-H., Lee, H.-E., Kim, Y., & Bien, Z. Z. (2008). A Steward Robot for Human-Friendly Human-Machine Interaction in a Smart House Environment. *IEEE Transactions on Automation Science and Engineering, 5(1), 21–25.*

<https://doi.org/10.1109/TASE.2007.911674>

Pigeon, S. (2005). *Use of Speech and Language Technology in Military Environments. Technology* (Vol. 323).

Protopapas, A., Tzakosta, M., Chalamandaris, A., & Tsiakoulis, P. (2012). IPLR: an online resource for Greek word-level and sublexical information. *Language Resources and Evaluation, 46(3), 449–459.* <https://doi.org/10.1007/s10579-010-9130-z>

R.K.Moore. (1994). Twenty things we still don't know about speech. In *Workshop on Progress and Prospects of speech Research and Technology.*

Rabiner, L. ., & Juang, B. H. (1986). An introduction to hidden Markov models. *IEEE ASSP Magazine, 3(January), 4–16.* <https://doi.org/10.1002/0471250953.bia03as18>

Remotely Operated Vehicle Committee of the Marine Technology Society. (n.d.).

Retrieved June 26, 2017, from http://www.rov.org/rov_overview.cfm

Rose, K., Eldridge, S., & Lyman, C. (2015). The internet of things: an overview. *Internet Society, (October), 53.* <https://doi.org/10.1017/CBO9781107415324.004>

Sagan, C., Sagan, L. S., & Drake, F. (1972). A Message from Earth. *Science, 175(4024).*

Retrieved from <http://science.sciencemag.org/content/175/4024/881>

- Samah, A., & Osman, A. A. (2015). Controlling Home Devices for Handicapped People via Voice Command Techniques.
- Seltzer, M. L., Ju, Y. C., Tashev, I., Wang, Y. Y., & Yu, D. (2011). In-car media search. *IEEE Signal Processing Magazine*, 28(4), 50–60.
<https://doi.org/10.1109/MSP.2011.941065>
- Senin, P. (2008). Dynamic Time Warping Algorithm Review. *Science*, 2007(December), 1–23. <https://doi.org/10.1109/IEMBS.2007.4353810>
- Singh, R., Warmuth, M. K., Raj, B., & Lamere, P. (2003). Classification with Free Energy at Raised Temperatures. *EuroSpeech*, (2), 1773–1776.
- Skaff, M. (2010). F-35 Lightning II Cockpit Vision. *SAE Int. J. Passeng. Cars – Electron. Electr. Syst.*, 3(2), 131–140. <https://doi.org/10.4271/2010-01-2330>
- Skype Translator Introduces Speech-to-Speech Translation in French and German – Translator. (n.d.). Retrieved May 3, 2017, from <https://blogs.msdn.microsoft.com/translation/2015/06/17/skype-translator-introduces-speech-to-speech-translation-in-french-and-german/>
- Speech at CMU. (n.d.). Retrieved May 14, 2017, from <http://www.speech.cs.cmu.edu/>
- Sphinx Knowledge Base Tool VERSION 3. (n.d.). Retrieved May 11, 2017, from <http://www.speech.cs.cmu.edu/tools/lmtool-new.html>
- Stallard, D., Prasad, R., Natarajan, P., Choi, F., Saleem, S., Meermeir, R., ... Devlin, J. (2009). The BBN TransTalk Speech-to-Speech Translation System. *English*, (June). <https://doi.org/10.5772/19405>
- Tsardoulas, E. G., Symeonidis, A. L., & Mitkas, P. A. (2015). An automatic speech detection architecture for social robot oral interaction. *Proceedings of the Audio Mostly*

2015 on Interaction With Sound. <https://doi.org/10.1145/2814895.2814931>

Use Raspberry Pi for Voice Control of Your Roomba | Make: (n.d.). Retrieved May 2, 2017, from <http://makezine.com/projects/use-raspberry-pi-for-voice-control/>

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2), 260–269. <https://doi.org/10.1109/TIT.1967.1054010>

Walker, W., Lamere, P., Kwok, P., Raj, B., Singh, R., Gouvea, E., ... Woelfel, J. (2004). Sphinx-4 : A Flexible Open Source Framework for Speech Recognition. *Sml*, (TR-2004-139), 1–9. <https://doi.org/10.1.1.91.4704>

Wang, Y. Y., Yu, D., Ju, Y. C., & Acero, A. (2008). An introduction to voice search. *IEEE Signal Processing Magazine*, 25(3), 29–38. <https://doi.org/10.1109/MSP.2008.918411>

What is Cortana? (n.d.). Retrieved May 3, 2017, from <https://support.microsoft.com/en-us/help/17214/windows-10-what-is>

Young, E. D. (2008). Neural representation of spectral and temporal information in speech. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 363(1493), 923–945. <https://doi.org/10.1098/rstb.2007.2151>

Yu, D. (2014). *Automatic Speech Recognition: A Deep Learning Approach*. Springer. <https://doi.org/10.1109/9780470546475>

ΒΟΥΛΗ ΤΩΝ ΕΛΛΗΝΩΝ. (n.d.). Retrieved May 15, 2017, from <http://www.parliament.gr/>

Διαδραστικά Σχολικά Βιβλία. (n.d.). Retrieved May 15, 2017, from <http://ebooks.edu.gr/new/>

Ελληνικός ορθογραφικός και γραμματικός έλεγχος. (n.d.). Retrieved May 14, 2017, from <http://www.elspell.gr/>

Μετάφραση Google. (n.d.). Retrieved May 15, 2017, from <https://translate.google.com/>

Πρόγραμμα σπουδών στη Σχεδίαση και Επεξεργασία Συστημάτων (Systems Engineering) - portal. (n.d.). Retrieved May 15, 2017, from <http://sse-tuc.edu.gr/web/guest/50>

Ιστοτόποι

<https://gitlab.sse.gr/fpantazoglou/omilia>

<https://goo.gl/9v3QqG>

<https://sourceforge.net/projects/cmusphinx/files/Acoustic%20and%20Language%20Models/Greek/>

<https://cmusphinx.github.io/>

<https://www.arduino.cc/>

<https://www.raspberrypi.org/>

<https://sourceforge.net/>

<http://stackoverflow.com/>

<https://github.com/>

<https://www.w3schools.com/>

<http://www.speech.cs.cmu.edu/>

<https://labs.oracle.com>

<http://www.merl.com/>

<http://rapp-project.eu/>

<http://cmusphinx.sourceforge.net/wiki/tutorial>

<http://www.darpa.mil>

<https://home.cern/>

<http://www.internetworldstats.com/>

<http://mcad.edu/>

<http://opencourses.gr/>

<http://cmusphinx.sourceforge.net>

<https://github.com/AdolfVonKleist/Phonetisaurus>

<http://www.hydro-lek.com/>

<http://www.fmctechnologies.com/en/SchillingRobotics.aspx>

<http://www.nato.int/>

<http://www.nhindustries.com>

<https://www.thalesgroup.com>

<https://wolfpaulus.com/embedded/raspberrypi2-sr/>

<https://goo.gl/w6WK6f>

<http://www.speech.cs.cmu.edu/databases/an4/index.html>

ΠΑΡΑΡΤΗΜΑ

Δημιουργία γλωσσικού μοντέλου

Κώδικας μακροεντολής breakLines (Jedit macro)

```
// Copyright Fotis Pantazoglou 2017 Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0)
SearchAndReplace.setSearchString("");
SearchAndReplace.setReplaceString("\n");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.setSearchFileSet(new CurrentBufferSet());
SearchAndReplace.replaceAll(view);
```

Κώδικας μακροεντολής CleanEmAll (Jedit macro)

```
// Copyright Fotis Pantazoglou 2017 Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0)
textArea.selectAll();
SearchAndReplace.setSearchString(" ");
SearchAndReplace.setReplaceString(" ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString(" ");
SearchAndReplace.setReplaceString(" ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
Registers.copy(textArea, '$');
SearchAndReplace.setSearchString("(");
SearchAndReplace.setReplaceString(" ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.setSearchFileSet(new CurrentBufferSet());
SearchAndReplace.replaceAll(view);
Registers.copy(textArea, '$');
SearchAndReplace.setSearchString(")");
SearchAndReplace.setReplaceString(" ");
SearchAndReplace.setBeanShellReplace(false);
```



```
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.setSearchFileSet(new CurrentBufferSet());
SearchAndReplace.replaceAll(view);
SearchAndReplace.setSearchString("«");
SearchAndReplace.setReplaceString(" ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.setSearchFileSet(new CurrentBufferSet());
SearchAndReplace.replaceAll(view);
Registers.copy(textArea,$);
SearchAndReplace.setSearchString("»");
SearchAndReplace.setReplaceString(" ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.setSearchFileSet(new CurrentBufferSet());
SearchAndReplace.replaceAll(view);
SearchAndReplace.setSearchString(",");
SearchAndReplace.setReplaceString(" ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.setSearchFileSet(new CurrentBufferSet());
SearchAndReplace.replaceAll(view);
SearchAndReplace.setSearchString("-");
SearchAndReplace.setReplaceString(" ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.setSearchFileSet(new CurrentBufferSet());
SearchAndReplace.replaceAll(view);
Registers.copy(textArea,$);
SearchAndReplace.setSearchString("-");
SearchAndReplace.setReplaceString(" ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.setSearchFileSet(new CurrentBufferSet());
SearchAndReplace.replaceAll(view);
```

```
SearchAndReplace.setSearchString("!");
SearchAndReplace.setReplaceString(" ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.setSearchFileSet(new CurrentBufferSet());
SearchAndReplace.replaceAll(view);
Registers.copy(textArea,$);
SearchAndReplace.setSearchString("—");
SearchAndReplace.setReplaceString(" ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.setSearchFileSet(new CurrentBufferSet());
SearchAndReplace.replaceAll(view);
Registers.copy(textArea,$);
SearchAndReplace.setSearchString("·");
SearchAndReplace.setReplaceString(" ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.setSearchFileSet(new CurrentBufferSet());
SearchAndReplace.replaceAll(view);
Registers.copy(textArea,$);
SearchAndReplace.setSearchString(";");
SearchAndReplace.setReplaceString(" ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.setSearchFileSet(new CurrentBufferSet());
SearchAndReplace.replaceAll(view);
textArea.selectAll();
Registers.copy(textArea,$);
SearchAndReplace.setSearchString("\\");
SearchAndReplace.setReplaceString(" ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replaceAll(view);
Registers.copy(textArea,$);
SearchAndReplace.setSearchString("\\");
SearchAndReplace.setReplaceString(" ");
```

```
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replaceAll(view);
Registers.copy(textArea,$);
SearchAndReplace.setSearchString("%");
SearchAndReplace.setReplaceString("τοις εκατό");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replaceAll(view);
textArea.selectAll();
textArea.toLowerCase();
textArea.selectAll();
SearchAndReplace.setSearchString(" ");
SearchAndReplace.setReplaceString(" ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
textArea.selectAll();
SearchAndReplace.setSearchString("... ");
SearchAndReplace.setReplaceString(" ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
textArea.selectAll();
SearchAndReplace.setSearchString("...");
SearchAndReplace.setReplaceString(" ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
textArea.selectAll();
SearchAndReplace.setSearchString("\");
SearchAndReplace.setReplaceString(" ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.setSearchFileSet(new CurrentBufferSet());
```

```
SearchAndReplace.replaceAll(view);
SearchAndReplace.setSearchString("*");
SearchAndReplace.setReplaceString(" ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.setSearchFileSet(new CurrentBufferSet());
SearchAndReplace.replaceAll(view);
```

Ενδεικτικό γλωσσικό μοντέλο test.lm

```
Copyright (c) 1996, Carnegie Mellon University, Cambridge University,
## Ronald Rosenfeld and Philip Clarkson
## Version 3, Copyright (c) 2006, Carnegie Mellon University
## Contributors includes Wen Xu, Ananlada Chotimongkol,
## David Huggins-Daines, Arthur Chan and Alan Black
===== This file was produced by the CMU-Cambridge =====
===== Statistical Language Modeling Toolkit =====
This is a 3-gram language model, based on a vocabulary of 84 words,
which begins "</s>", "<s>", "uber"...
This is a CLOSED-vocabulary model
(OOVs eliminated from training data and are forbidden in test data)
Good-Turing discounting was applied.
1-gram frequency of frequency : 64
2-gram frequency of frequency : 101 8 0 0 0 0
3-gram frequency of frequency : 109 4 0 0 0 0
1-gram discounting ratios : 0.98
2-gram discounting ratios : 0.00
3-gram discounting ratios : 0.00
This file is in the ARPA-standard format introduced by Doug Paul.
p(wd3|wd1,wd2)= if(trigram exists)      p_3(wd1,wd2,wd3)
                else if(bigram w1,w2 exists) bo_wt_2(w1,w2)*p(wd3|wd2)
                else                        p(wd3|w2)
p(wd2|wd1)= if(bigram exists) p_2(wd1,wd2)
                else          bo_wt_1(wd1)*p_1(wd2)
All probs and back-off weights (bo_wt) are given in log10 form.
Data formats:
Beginning of data mark: \data\
ngram 1=nr      # number of 1-grams
ngram 2=nr      # number of 2-grams
ngram 3=nr      # number of 3-grams
\1-grams:
p_1  wd_1 bo_wt_1
\2-grams:
p_2  wd_1 wd_2 bo_wt_2
```

\3-grams:

p_3 wd_1 wd_2 wd_3

end of data mark: \end\

\data\

ngram 1=84

ngram 2=109

ngram 3=113

\1-grams:

-1.7672 </s> -0.4658

-1.5911 <s> 0.0111

-1.5911 uber 0.0388

-2.0749 ένωσης 0.0113

-2.0749 αδείας 0.0037

-1.7672 αλλά 0.0189

-2.0749 αναφέρουν 0.0037

-2.0749 από 0.0075

-1.7672 αρχή -0.5592

-2.0749 βάζει 0.0037

-2.0749 γενικού 0.0075

-2.0749 γενικό 0.0075

-1.5911 για 0.0111

-2.0749 δίκαιο 0.0429

-1.5911 δεν 0.0150

-2.0749 δεσμεύσεις 0.0037

-2.0749 διέπεται 0.0037

-2.0749 διασφαλίζει 0.0075

-2.0749 δικαστήριο 0.0075

-2.0749 δράση 0.0429

-2.0749 δραστηριότητά 0.0429

-2.0749 δυνατό 0.0037

-1.7672 είναι 0.0074

-2.0749 εε 0.0037

-2.0749 εισήγηση 0.0037

-1.7672 εισαγγελέα 0.0189

-2.0749 εισαγγελέας 0.0037

-1.7672 ελεύθερης -0.4696

-2.0749 εμπίπτει 0.0037

-2.0749 εντός 0.0429

-2.0749 επιβατικές 0.0037

-2.0749 επισημαίνει 0.0037

-2.0749 ευρωπαϊκού 0.0037

-2.0749 ευρωπαϊκό 0.0037

-1.4661 η 0.0188

-2.0749 ηλεκτρονικής 0.0037

-2.0749 θεωρώντας 0.0075
 -2.0749 ισχύει 0.0037
 -2.0749 ιχ 0.0113
 -2.0749 και 0.0037
 -2.0749 κατά 0.0037
 -2.0749 κατοχή 0.0037
 -1.7672 κοινωνίας -0.5592
 -2.0749 μίσθωση 0.0037
 -2.0749 μεταφοράς 0.0075
 -2.0749 μεταφορές 0.0037
 -2.0749 μεταφορών 0.0037
 -2.0749 μόνον 0.0037
 -2.0749 νέα 0.0075
 -2.0749 να 0.0037
 -2.0749 ο 0.0037
 -2.0749 οποία 0.0037
 -1.7672 παροχής 0.0112
 -2.0749 πλατφόρμα 0.0113
 -2.0749 πλατφόρμας 0.0113
 -1.7672 πληροφορίας 0.0112
 -2.0749 προς 0.0075
 -2.0749 πρόταση 0.0037
 -2.0749 στη 0.0037
 -2.0749 στις 0.0037
 -2.0749 τέτοια 0.0000
 -2.0749 τα 0.0037
 -2.0749 ταξι 0.0075
 -2.0749 τη 0.0037
 -1.7672 την 0.0112
 -1.0268 της -0.3641
 -2.0749 τις 0.0037
 -1.7672 το 0.0074
 -2.0749 τομέα 0.0037
 -2.0749 τον 0.0037
 -2.0749 του 0.0037
 -2.0749 των 0.0037
 -2.0749 υπέρ 0.0429
 -2.0749 υπηρεσίες 0.0429
 -1.7672 υπηρεσιών 0.0512
 -2.0749 υποχρεωτική 0.0075
 -2.0749 φρένο 0.0037
 -2.0749 χαρακτηριστεί 0.0037
 -2.0749 ψηφιακών 0.0075
 -2.0749 ως 0.0037

-2.0749 όπου 0.0151
 -2.0749 όπως 0.0037
 -1.7672 ότι -0.4658
 -2.0749 όχι 0.0037
 \2-grams:
 -0.1761 </s> <s> 0.0000
 -99.9990 <s> κατά 0.0000
 -99.9990 <s> ο 0.0000
 -99.9990 <s> φρένο 0.0000
 -99.9990 uber για 0.0000
 -99.9990 uber δεν 0.0000
 -99.9990 uber η 0.0000
 -99.9990 ένωσης για 0.0000
 -99.9990 αδείας μεταφοράς 0.0000
 -99.9990 αλλά η 0.0000
 -99.9990 αλλά και 0.0000
 -99.9990 αναφέρουν τα 0.0000
 -99.9990 από την 0.0000
 -0.1249 αρχή της -0.3632
 -99.9990 βάζει πρόταση 0.0000
 -99.9990 γενικού εισαγγελέα 0.0000
 -99.9990 γενικό εισαγγελέα 0.0000
 -99.9990 για επιβατικές 0.0000
 -99.9990 για τη 0.0000
 -99.9990 για τις 0.0000
 -99.9990 δίκαιο της 0.0000
 -99.9990 δεν διέπεται 0.0000
 -99.9990 δεν είναι 0.0000
 -99.9990 δεν ισχύει 0.0000
 -99.9990 δεσμεύσεις τομέα 0.0000
 -99.9990 διέπεται από 0.0000
 -99.9990 διασφαλίζει το 0.0000
 -99.9990 δικαστήριο </s> 0.4771
 -99.9990 δράση της 0.0000
 -99.9990 δραστηριότητά της 0.0000
 -99.9990 δυνατό να 0.0000
 -99.9990 είναι δυνατό 0.0000
 -99.9990 είναι υποχρεωτική 0.0000
 -99.9990 σε βάζει 0.0000
 -99.9990 εισήγηση του 0.0000
 -99.9990 εισαγγελέα η 0.0000
 -99.9990 εισαγγελέα προς 0.0000
 -99.9990 εισαγγελέας επισημαίνει 0.0000
 -0.1761 ελεύθερης παροχής 0.0000

-99.9990 εμπίπτει στις 0.0000
-99.9990 εντός της 0.0000
-99.9990 επιβατικές μεταφορές 0.0000
-99.9990 επισημαίνει όπως 0.0000
-99.9990 ευρωπαϊκού γενικού 0.0000
-99.9990 ευρωπαϊκό δικαστήριο 0.0000
-99.9990 η αρχή 0.6021
-99.9990 η δραστηριότητά 0.0000
-99.9990 η κατοχή 0.0000
-99.9990 η πλατφόρμα 0.0000
-99.9990 ηλεκτρονικής πλατφόρμας 0.0000
-99.9990 θεωρώντας ότι 0.4771
-99.9990 ισχύει υπέρ 0.0000
-99.9990 ιχ για 0.0000
-99.9990 και ιχ 0.0000
-99.9990 κατά τον 0.0000
-99.9990 κατοχή αδειάς 0.0000
-0.1249 κοινωνίας της -0.4046
-99.9990 μίσθωση όχι 0.0000
-99.9990 μεταφορές είναι 0.0000
-99.9990 μεταφορές εντός 0.0000
-99.9990 μεταφορών όπου 0.0000
-99.9990 μόνον ταξί 0.0000
-99.9990 νέα ότι 0.4771
-99.9990 να χαρακτηριστεί 0.0000
-99.9990 ο εισαγγελέας 0.0000
-99.9990 οποία διασφαλίζει 0.0000
-99.9990 παροχής υπηρεσιών 0.0000
-99.9990 παροχής ψηφιακών 0.0000
-99.9990 πλατφόρμα uber 0.0000
-99.9990 πλατφόρμας uber 0.0000
-99.9990 πληροφορίας αλλά 0.0000
-99.9990 πληροφορίας θεωρώντας 0.0000
-99.9990 προς το 0.0000
-99.9990 πρόταση εισήγηση 0.0000
-99.9990 στη δράση 0.0000
-99.9990 στις δεσμεύσεις 0.0000
-99.9990 τα νέα 0.0000
-99.9990 ταξί αλλά 0.0000
-99.9990 τη μίσθωση 0.0000
-99.9990 την αρχή 0.6021
-99.9990 την οποία 0.0000
-99.9990 της uber 0.0000
-99.9990 της ένωσης 0.0000

-99.9990 της εε 0.0000
-0.6368 της ελεύθερης 0.0000
-99.9990 της εμπίπτει 0.0000
-99.9990 της ηλεκτρονικής 0.0000
-0.6368 της κοινωνίας 0.1249
-0.8129 της πληροφορίας 0.0000
-99.9990 τις υπηρεσίες 0.0000
-99.9990 το δίκαιο 0.0000
-99.9990 το ευρωπαϊκό 0.0000
-99.9990 τομέα των 0.0000
-99.9990 τον γενικό 0.0000
-99.9990 του ευρωπαίου 0.0000
-99.9990 των μεταφορών 0.0000
-99.9990 υπέρ της 0.0000
-99.9990 υπηρεσίες της 0.1139
-99.9990 υπηρεσιών την 0.0000
-99.9990 υπηρεσιών της 0.1139
-99.9990 υποχρεωτική </s> 0.4771
-99.9990 φρένο στη 0.0000
-99.9990 χαρακτηριστεί ως 0.0000
-99.9990 ψηφιακών υπηρεσιών 0.0000
-99.9990 ως τέτοια 0.0075
-99.9990 όπου η 0.0000
-99.9990 όπως αναφέρουν 0.0000
-0.1761 ότι δεν 0.0000
-99.9990 όχι μόνον 0.0000
\\3-grams:
-99.9990 </s> <s> κατά
-99.9990 </s> <s> ο
-99.9990 <s> κατά τον
-99.9990 <s> ο εισαγγελέας
-99.9990 <s> φρένο στη
-99.9990 uber για τη
-99.9990 uber δεν διέπεται
-99.9990 uber η αρχή
-99.9990 ένωσης για τις
-99.9990 αδείας μεταφοράς είναι
-99.9990 αλλά η δραστηριότητά
-99.9990 αλλά και ιχ
-99.9990 αναφέρουν τα νέα
-99.9990 από την αρχή
-0.1761 αρχή της ελεύθερης
-99.9990 βάζει πρόταση εισήγηση
-99.9990 γενικού εισαγγελέα προς

-99.9990 γενικό εισαγγελέα η
-99.9990 για επιβατικές μεταφορές
-99.9990 για τη μίσθωση
-99.9990 για τις υπηρεσίες
-99.9990 δικαιο της ένωσης
-99.9990 δεν διέπεται από
-99.9990 δεν είναι δυνατό
-99.9990 δεν ισχύει υπέρ
-99.9990 δεσμεύσεις τομέα των
-99.9990 διέπεται από την
-99.9990 διασφαλίζει το δικαιο
-99.9990 δικαστήριο </s> <s>
-99.9990 δράση της ηλεκτρονικής
-99.9990 δραστηριότητά της εμπίπτει
-99.9990 δυνατό να χαρακτηριστεί
-99.9990 είναι δυνατό να
-99.9990 είναι υποχρεωτική </s>
-99.9990 εε βάζει πρόταση
-99.9990 εισήγηση του ευρωπαϊού
-99.9990 εισαγγελέα η πλατφόρμα
-99.9990 εισαγγελέα προς το
-99.9990 εισαγγελέας επισημαίνει όπως
-99.9990 ελεύθερης παροχής υπηρεσιών
-99.9990 ελεύθερης παροχής ψηφιακών
-99.9990 εμπίπτει στις δεσμεύσεις
-99.9990 εντός της εε
-99.9990 επιβατικές μεταφορές εντός
-99.9990 επισημαίνει όπως αναφέρουν
-99.9990 ευρωπαϊού γενικού εισαγγελέα
-99.9990 ευρωπαϊκό δικαστήριο </s>
-99.9990 η αρχή της
-99.9990 η δραστηριότητά της
-99.9990 η κατοχή αδείας
-99.9990 η πλατφόρμα uber
-99.9990 ηλεκτρονικής πλατφόρμας uber
-99.9990 θεωρώντας ότι δεν
-99.9990 ισχύει υπέρ της
-99.9990 ιχ για επιβατικές
-99.9990 και ιχ για
-99.9990 κατά τον γενικό
-99.9990 κατοχή αδείας μεταφοράς
-0.1761 κοινωνίας της πληροφορίας
-99.9990 μίσθωση όχι μόνον
-99.9990 μεταφοράς είναι υποχρεωτική

-99.9990 μεταφορές εντός της
-99.9990 μεταφορών όπου η
-99.9990 μόνον ταξί αλλά
-99.9990 νέα ότι δεν
-99.9990 να χαρακτηριστεί ως
-99.9990 ο εισαγγελέας επισημαίνει
-99.9990 οποία διασφαλίζει το
-99.9990 παροχής υπηρεσιών την
-99.9990 παροχής ψηφιακών υπηρεσιών
-99.9990 πλατφόρμα uber δεν
-99.9990 πλατφόρμα uber για
-99.9990 πληροφορίας αλλά η
-99.9990 πληροφορίας θεωρώντας ότι
-99.9990 προς το ευρωπαϊκό
-99.9990 πρόταση εισήγηση του
-99.9990 στη δράση της
-99.9990 στις δεσμεύσεις τομέα
-99.9990 τα νέα ότι
-99.9990 ταξί αλλά και
-99.9990 τη μίσθωση όχι
-99.9990 την αρχή της
-99.9990 την οποία διασφαλίζει
-99.9990 της uber η
-99.9990 της ένωσης για
-99.9990 της σε βάζει
-0.1761 της ελεύθερης παροχής
-99.9990 της εμπίπτει στις
-99.9990 της ηλεκτρονικής πλατφόρμας
-0.1761 της κοινωνίας της
-99.9990 της πληροφορίας αλλά
-99.9990 της πληροφορίας θεωρώντας
-99.9990 τις υπηρεσίες της
-99.9990 το δικαίο της
-99.9990 το ευρωπαϊκό δικαστήριο
-99.9990 τομέα των μεταφορών
-99.9990 τον γενικό εισαγγελέα
-99.9990 του ευρωπαίου γενικού
-99.9990 των μεταφορών όπου
-99.9990 υπέρ της uber
-99.9990 υπηρεσίες της κοινωνίας
-99.9990 υπηρεσιών την οποία
-99.9990 υπηρεσιών της κοινωνίας
-99.9990 υποχρεωτική </s> <s>
-99.9990 φρένο στη δράση

```
-99.9990 χαρακτηριστεί ως τέτοια  
-99.9990 ψηφιακών υπηρεσιών της  
-99.9990 ως τέτοια </s>  
-99.9990 όπου η κατοχή  
-99.9990 όπως αναφέρονται  
-99.9990 ότι δεν είναι  
-99.9990 ότι δεν ισχύει  
-99.9990 όχι μόνον τάζει  
\end\
```

Δημιουργία φωνητικού λεξικού

Κώδικας μακροεντολής create_phoneme (Jedit macro)

// Copyright Fotis Pantazoglou 2017 Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0)

```
textArea.selectAll();  
textArea.toLowerCase();  
textArea.selectAll();  
SearchAndReplace.setSearchString("ου");  
SearchAndReplace.setReplaceString("UW");  
SearchAndReplace.setBeanShellReplace(false);  
SearchAndReplace.setWholeWord(false);  
SearchAndReplace.setIgnoreCase(false);  
SearchAndReplace.setRegexp(false);  
SearchAndReplace.replace(view);  
SearchAndReplace.setSearchString("ού");  
SearchAndReplace.setReplaceString("UW");  
SearchAndReplace.setBeanShellReplace(false);  
SearchAndReplace.setWholeWord(false);  
SearchAndReplace.setIgnoreCase(false);  
SearchAndReplace.setRegexp(false);  
SearchAndReplace.replace(view);  
SearchAndReplace.setSearchString("σμπ");  
SearchAndReplace.setReplaceString("ZB");  
SearchAndReplace.setBeanShellReplace(false);  
SearchAndReplace.setWholeWord(false);  
SearchAndReplace.setIgnoreCase(false);  
SearchAndReplace.setRegexp(false);  
SearchAndReplace.replace(view);  
SearchAndReplace.setSearchString("σντ");  
SearchAndReplace.setReplaceString("ZD");  
SearchAndReplace.setBeanShellReplace(false);  
SearchAndReplace.setWholeWord(false);  
SearchAndReplace.setIgnoreCase(false);  
SearchAndReplace.setRegexp(false);  
SearchAndReplace.replace(view);  
SearchAndReplace.setSearchString("σγ");
```



```
SearchAndReplace.setReplaceString("ZW");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("σβ");
SearchAndReplace.setReplaceString("ZV");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("σδ");
SearchAndReplace.setReplaceString("ZDH");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("σμ");
SearchAndReplace.setReplaceString("ZM");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("σν");
SearchAndReplace.setReplaceString("ZN");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("σλ");
SearchAndReplace.setReplaceString("ZL");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("σρ");
SearchAndReplace.setReplaceString("ZR");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
```

```
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("μπ");
SearchAndReplace.setReplaceString("B");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ντ");
SearchAndReplace.setReplaceString("D");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("γκ");
SearchAndReplace.setReplaceString("G");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("γγ");
SearchAndReplace.setReplaceString("G");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("τσ");
SearchAndReplace.setReplaceString("CH");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("τζ");
SearchAndReplace.setReplaceString("JH");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("σσ");
SearchAndReplace.setReplaceString("S");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
```

```
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("κκ");
SearchAndReplace.setReplaceString("Κ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ββ");
SearchAndReplace.setReplaceString("Β");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("λλ");
SearchAndReplace.setReplaceString("Λ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("μμ");
SearchAndReplace.setReplaceString("Μ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("νν");
SearchAndReplace.setReplaceString("Ν");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ππ");
SearchAndReplace.setReplaceString("Π");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ρρ");
SearchAndReplace.setReplaceString("Ρ");
```

```
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ττ");
SearchAndReplace.setReplaceString("T");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("α");
SearchAndReplace.setReplaceString("EH");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("α");
SearchAndReplace.setReplaceString("EH");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ε");
SearchAndReplace.setReplaceString("IH");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ε");
SearchAndReplace.setReplaceString("IH");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ο");
SearchAndReplace.setReplaceString("IH");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
```

```
SearchAndReplace.setSearchString("oi");
SearchAndReplace.setReplaceString("IH");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("vi");
SearchAndReplace.setReplaceString("IH");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("vi");
SearchAndReplace.setReplaceString("IH");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("av");
SearchAndReplace.setReplaceString("αβ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("αυ");
SearchAndReplace.setReplaceString("αβ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ευ");
SearchAndReplace.setReplaceString("εβ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ευ");
SearchAndReplace.setReplaceString("εβ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
```

```
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("αυ");
SearchAndReplace.setReplaceString("αφ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("αύ");
SearchAndReplace.setReplaceString("άφ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ευ");
SearchAndReplace.setReplaceString("εφ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("εύ");
SearchAndReplace.setReplaceString("έφ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("Α");
SearchAndReplace.setReplaceString("ΑΑ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("α");
SearchAndReplace.setReplaceString("ΑΑ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ά");
SearchAndReplace.setReplaceString("ΑΑ");
SearchAndReplace.setBeanShellReplace(false);
```



```
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("β");
SearchAndReplace.setReplaceString("V");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("γ");
SearchAndReplace.setReplaceString("W");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("δ");
SearchAndReplace.setReplaceString("DH");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ε");
SearchAndReplace.setReplaceString("EH");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ε");
SearchAndReplace.setReplaceString("EH");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ζ");
SearchAndReplace.setReplaceString("Z");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("η");
```

```
SearchAndReplace.setReplaceString("IH");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ή");
SearchAndReplace.setReplaceString("IH");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("θ");
SearchAndReplace.setReplaceString("TH");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ι");
SearchAndReplace.setReplaceString("IH");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ι");
SearchAndReplace.setReplaceString("IH");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ι");
SearchAndReplace.setReplaceString("IH");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ι");
SearchAndReplace.setReplaceString("IH");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
```

```
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("κ");
SearchAndReplace.setReplaceString("Κ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("λ");
SearchAndReplace.setReplaceString("Λ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("μ");
SearchAndReplace.setReplaceString("Μ");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ν");
SearchAndReplace.setReplaceString("Ν");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ξ");
SearchAndReplace.setReplaceString("KS");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ο");
SearchAndReplace.setReplaceString("OW");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ό");
SearchAndReplace.setReplaceString("OW");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
```

```
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("π");
SearchAndReplace.setReplaceString("P");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ρ");
SearchAndReplace.setReplaceString("R");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("σ");
SearchAndReplace.setReplaceString("S");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("τ");
SearchAndReplace.setReplaceString("T");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("υ");
SearchAndReplace.setReplaceString("IH");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ϋ");
SearchAndReplace.setReplaceString("IH");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ϋ");
SearchAndReplace.setReplaceString("IH");
```

```
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ϔ");
SearchAndReplace.setReplaceString("IH");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("φ");
SearchAndReplace.setReplaceString("F");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("χ");
SearchAndReplace.setReplaceString("HH");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ψ");
SearchAndReplace.setReplaceString("PS");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ω");
SearchAndReplace.setReplaceString("OW");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
SearchAndReplace.setSearchString("ώ");
SearchAndReplace.setReplaceString("OW");
SearchAndReplace.setBeanShellReplace(false);
SearchAndReplace.setWholeWord(false);
SearchAndReplace.setIgnoreCase(false);
SearchAndReplace.setRegexp(false);
SearchAndReplace.replace(view);
```

```
SearchAndReplace.setSearchString("ς");  
SearchAndReplace.setReplaceString("S");  
SearchAndReplace.setBeanShellReplace(false);  
SearchAndReplace.setWholeWord(false);  
SearchAndReplace.setIgnoreCase(false);  
SearchAndReplace.setRegexp(false);  
SearchAndReplace.replace(view);
```

Δημιουργία ακουστικού μοντέλου

Κείμενο προς ηχογράφηση

ξεκίνα πορεία ευθεία σιγά
στρίψε δεξιά στρίψε αριστερά σιγά σιγά
σταμάτα ξεκίνα συνέχισε κάνε πίσω
αύξησε ταχύτητα μείωσε ταχύτητα
εκατόν ογδόντα μοίρες
κάνε στροφή σαράντα πέντε μοιρών
πάρε δείγμα πιάσε αντικείμενο
άνοιξε κλείσε δίκτυο κλείσε δαγκάνα
πορεία εκατόν είκοσι μοίρες
πιο ήρεμα εξήντα μοίρες
ακόμη λίγο ανέβασε ένταση
μείωσε ένταση άνοιξε βραχίονα
κλείσε βραχίονα δεξιό αριστερό
εμπρός πίσω δεξιά αριστερά ευθεία
κάνε κύκλο ενενήντα μοίρες άνοιξε φώτα
άφησε αντικείμενο άνοιξε δαγκάνα
πορεία διακόσιες εβδομήντα μοίρες
ξεκίνα ένα σταμάτα ένα
ένα πιο πολύ ένα πιο λίγο
ξεκίνα δύο σταμάτα δύο
δύο πιο πολύ δύο πιο λίγο
ξεκίνα τρία σταμάτα τρία
τρία πιο πολύ τρία πιο λίγο
ξεκίνα τέσσερα σταμάτα τέσσερα
τέσσερα πιο πολύ τέσσερα πιο λίγο
ξεκίνα πέντε σταμάτα πέντε
πέντε πιο πολύ πέντε πιο λίγο
τριακόσιες εξήντα μοίρες κλείσε φώτα
πάνω κάτω δεξιά αριστερά σιγά γρήγορα
δείγμα αντικείμενο άνοιξε κλείσε

sphinx_train.cfg

```
$CFG_VERBOSE = 1;           # Determines how much goes to the screen.

# These are filled in at configuration time
$CFG_DB_NAME = "an4";
# Experiment name, will be used to name model files and log files
$CFG_EXPTNAME = "$CFG_DB_NAME";

# Directory containing SphinxTrain binaries
$CFG_BASE_DIR = "/home/fantom/tutorial/an4";
$CFG_SPHINXTRAIN_DIR = "/usr/local/lib/sphinxtrain";
$CFG_BIN_DIR = "/usr/local/libexec/sphinxtrain";
$CFG_SCRIPT_DIR = "/usr/local/lib/sphinxtrain/scripts";

# Audio waveform and feature file information
$CFG_WAVFILES_DIR = "$CFG_BASE_DIR/wav/fort";
$CFG_WAVFILE_EXTENSION = 'wav';
$CFG_WAVFILE_TYPE = 'mswav'; # one of nist, mswav, raw
$CFG_FEATFILES_DIR = "$CFG_BASE_DIR/feat";
$CFG_FEATFILE_EXTENSION = 'mfc';

# Feature extraction parameters
$CFG_WAVFILE_SRATE = 16000.0;
$CFG_NUM_FILT = 25; # For wideband speech it's 25, for telephone 8khz reasonable value is 15
$CFG_LO_FILT = 130; # For telephone 8kHz speech value is 200
$CFG_HI_FILT = 6800; # For telephone 8kHz speech value is 3500
$CFG_TRANSFORM = "dct"; # Previously legacy transform is used, but dct is more accurate
$CFG_LIFTER = "22"; # Cepstrum lifter is smoothing to improve recognition
$CFG_VECTOR_LENGTH = 13; # 13 is usually enough

$CFG_MIN_ITERATIONS = 1; # BW Iterate at least this many times
$CFG_MAX_ITERATIONS = 10; # BW Don't iterate more than this, somethings likely wrong.

# (none/max) Type of AGC to apply to input files
$CFG_AGC = 'none';
# (current/none) Type of cepstral mean subtraction/normalization
# to apply to input files
$CFG_CMN = 'batch';
# (yes/no) Normalize variance of input files to 1.0
$CFG_VARNORM = 'no';
# (yes/no) Train full covariance matrices
$CFG_FULLVAR = 'no';
```

```
# (yes/no) Use diagonals only of full covariance matrices for
# Forward-Backward evaluation (recommended if CFG_FULLVAR is yes)
$CFG_DIAGFULL = 'no';

# (yes/no) Perform vocal tract length normalization in training. This
# will result in a "normalized" model which requires VTLN to be done
# during decoding as well.
$CFG_VTLN = 'no';
# Starting warp factor for VTLN
$CFG_VTLN_START = 0.80;
# Ending warp factor for VTLN
$CFG_VTLN_END = 1.40;
# Step size of warping factors
$CFG_VTLN_STEP = 0.05;

# Directory to write queue manager logs to
$CFG_QMGR_DIR = "$CFG_BASE_DIR/qmanager";
# Directory to write training logs to
$CFG_LOG_DIR = "$CFG_BASE_DIR/logdir";
# Directory for re-estimation counts
$CFG_BWACCUM_DIR = "$CFG_BASE_DIR/bwaccumdir";
# Directory to write model parameter files to
$CFG_MODEL_DIR = "$CFG_BASE_DIR/model_parameters";

# Directory containing transcripts and control files for
# speaker-adaptive training
$CFG_LIST_DIR = "$CFG_BASE_DIR/etc";

# Decoding variables for MMIE training
$CFG_LANGUAGEWEIGHT = "11.5";
$CFG_BEAMWIDTH = "1e-100";
$CFG_WORDBEAM = "1e-80";
$CFG_LANGUAGEMODEL = "$CFG_LIST_DIR/$CFG_DB_NAME.lm.DMP";
$CFG_WORDPENALTY = "0.2";

# Lattice pruning variables
$CFG_ABEAM = "1e-50";
$CFG_NBEAM = "1e-10";
$CFG_PRUNED_DENLAT_DIR = "$CFG_BASE_DIR/pruned_denlat";

# MMIE training related variables
$CFG_MMIE = "no";
$CFG_MMIE_MAX_ITERATIONS = 5;
$CFG_LATTICE_DIR = "$CFG_BASE_DIR/lattice";
```

```
$CFG_MMIE_TYPE = "rand"; # Valid values are "rand", "best" or "ci"
$CFG_MMIE_CONSTE = "3.0";
$CFG_NUMLAT_DIR = "$CFG_BASE_DIR/numlat";
$CFG_DENLAT_DIR = "$CFG_BASE_DIR/denlat";

# Variables used in main training of models
$CFG_DICTIONARY = "$CFG_LIST_DIR/el-gr.dic";
$CFG_RAWPHONEFILE = "$CFG_LIST_DIR/el-gr.phone";
$CFG_FILLERDICT = "$CFG_LIST_DIR/el-gr.filler";
$CFG_LISTOFFILES = "$CFG_LIST_DIR/fot_train.fileids";
$CFG_TRANSCRIPTFILE = "$CFG_LIST_DIR/fot_train.transcription";
$CFG_FEATPARAMS = "$CFG_LIST_DIR/feat.params";

# Variables used in characterizing models

$CFG_HMM_TYPE = 'cont.'; # Sphinx 4, PocketSphinx
#$CFG_HMM_TYPE = 'semi.'; # PocketSphinx
#$CFG_HMM_TYPE = 'ptm.'; # PocketSphinx (larger data sets)

if (($CFG_HMM_TYPE ne ".semi.")
    and ($CFG_HMM_TYPE ne ".ptm.")
    and ($CFG_HMM_TYPE ne ".cont.")) {
    die "Please choose one CFG_HMM_TYPE out of '.cont.', '.ptm.', or '.semi.', " .
        "currently $CFG_HMM_TYPE\n";
}

# This configuration is fastest and best for most acoustic models in
# PocketSphinx and Sphinx-III. See below for Sphinx-II.
$CFG_STATESPERHMM = 3;
$CFG_SKIPSTATE = 'no';

if ($CFG_HMM_TYPE eq '.semi.') {
    $CFG_DIRLABEL = 'semi';
    # Four stream features for PocketSphinx
    $CFG_FEATURE = "s2_4x";
    $CFG_NUM_STREAMS = 4;
    $CFG_INITIAL_NUM_DENSITIES = 256;
    $CFG_FINAL_NUM_DENSITIES = 256;
    die "For semi continuous models, the initial and final models have the same density"
        if ($CFG_INITIAL_NUM_DENSITIES != $CFG_FINAL_NUM_DENSITIES);
} elsif ($CFG_HMM_TYPE eq '.ptm.') {
    $CFG_DIRLABEL = 'ptm';
    # Four stream features for PocketSphinx
    $CFG_FEATURE = "s2_4x";
```

```
$CFG_NUM_STREAMS = 4;
$CFG_INITIAL_NUM_DENSITIES = 64;
$CFG_FINAL_NUM_DENSITIES = 64;
die "For phonetically tied models, the initial and final models have the same density"
  if ($CFG_INITIAL_NUM_DENSITIES != $CFG_FINAL_NUM_DENSITIES);
} elsif ($CFG_HMM_TYPE eq 'cont.') {
  $CFG_DIRLABEL = 'cont';
# Single stream features - Sphinx 3
$CFG_FEATURE = "1s_c_d_dd";
$CFG_NUM_STREAMS = 1;
$CFG_INITIAL_NUM_DENSITIES = 1;
$CFG_FINAL_NUM_DENSITIES = 8;
die "The initial has to be less than the final number of densities"
  if ($CFG_INITIAL_NUM_DENSITIES > $CFG_FINAL_NUM_DENSITIES);
}

# Number of top gaussians to score a frame. A little bit less accurate computations
# make training significantly faster. Uncomment to apply this during the training
# For good accuracy make sure you are using the same setting in decoder
# In theory this can be different for various training stages. For example 4 for
# CI stage and 16 for CD stage
# $CFG_CI_TOPN = 4;
# $CFG_CD_TOPN = 16;

# (yes/no) Train multiple-gaussian context-independent models (useful
# for alignment, use 'no' otherwise) in the models created
# specifically for forced alignment
$CFG_FALIGN_CI_MGAU = 'no';
# (yes/no) Train multiple-gaussian context-independent models (useful
# for alignment, use 'no' otherwise)
$CFG_CI_MGAU = 'no';
# (yes/no) Train context-dependent models
$CFG_CD_TRAIN = 'no';
# Number of tied states (senones) to create in decision-tree clustering
$CFG_N_TIED_STATES = 200;
# How many parts to run Forward-Backward estimation in
$CFG_NPART = 1;

# (yes/no) Train a single decision tree for all phones (actually one
# per state) (useful for grapheme-based models, use 'no' otherwise)
$CFG_CROSS_PHONE_TREES = 'no';

# Use force-aligned transcripts (if available) as input to training
$CFG_FORCEDALIGN = 'no';
```

```
# Use a specific set of models for force alignment. If not defined,
# context-independent models for the current experiment will be used.
$CFG_FORCE_ALIGN_MODELDIR = "$CFG_MODEL_DIR/$CFG_EXPTNAME.falign_ci_$CFG_DIRLABEL";

# Use a specific dictionary and filler dictionary for force alignment.
# If these are not defined, a dictionary and filler dictionary will be
# created from $CFG_DICTIONARY and $CFG_FILLERDICT, with noise words
# removed from the filler dictionary and added to the dictionary (this
# is because the force alignment is not very good at inserting them)

# $CFG_FORCE_ALIGN_DICTIONARY = "$ST::CFG_BASE_DIR/falignout$ST::CFG_EXPTNAME.falign.dict";
# $CFG_FORCE_ALIGN_FILLERDICT = "$ST::CFG_BASE_DIR/falignout/$ST::CFG_EXPTNAME.falign.fdict";

# Use a particular beam width for force alignment. The wider
# (i.e. smaller numerically) the beam, the fewer sentences will be
# rejected for bad alignment.
$CFG_FORCE_ALIGN_BEAM = 1e-60;

# Calculate an LDA/MLLT transform?
$CFG_LDA_MLLT = 'no';
# Dimensionality of LDA/MLLT output
$CFG_LDA_DIMENSION = 29;

# This is actually just a difference in log space (it doesn't make
# sense otherwise, because different feature parameters have very
# different likelihoods)
$CFG_CONVERGENCE_RATIO = 0.1;

# Queue::POSIX for multiple CPUs on a local machine
# Queue::PBS to use a PBS/TORQUE queue
$CFG_QUEUE_TYPE = "Queue::POSIX";

# Name of queue to use for PBS/TORQUE
$CFG_QUEUE_NAME = "workq";

# (yes/no) Build questions for decision tree clustering automatically
$CFG_MAKE_QUESTS = "yes";
# If CFG_MAKE_QUESTS is yes, questions are written to this file.
# If CFG_MAKE_QUESTS is no, questions are read from this file.
$CFG_QUESTION_SET = "${CFG_BASE_DIR}/model_architecture/${CFG_EXPTNAME}.tree_questions";
# $CFG_QUESTION_SET = "${CFG_BASE_DIR}/linguistic_questions";

$CFG_CP_OPERATION = "${CFG_BASE_DIR}/model_architecture/${CFG_EXPTNAME}.cpmeanvar";
```

```
# Configuration for grapheme-to-phoneme model
$CFG_G2P_MODEL='no';

# Configuration script for sphinx decoder

# Variables starting with $DEC_CFG_ refer to decoder specific
# arguments, those starting with $CFG_ refer to trainer arguments,
# some of them also used by the decoder.

$DEC_CFG_VERBOSE = 1;           # Determines how much goes to the screen.

# These are filled in at configuration time

# Name of the decoding script to use (psdecode.pl or s3decode.pl, probably)
$DEC_CFG_SCRIPT = 'psdecode.pl';

$DEC_CFG_EXPTNAME = "$CFG_EXPTNAME";
$DEC_CFG_JOBNAME = "$CFG_EXPTNAME"._job";

# Models to use.
$DEC_CFG_MODEL_NAME = "$CFG_EXPTNAME.ci_${CFG_DIRLABEL}";

$DEC_CFG_FEATFILES_DIR = "$CFG_BASE_DIR/feat";
$DEC_CFG_FEATFILE_EXTENSION = '.mfc';
$DEC_CFG_AGC = $CFG_AGC;
$DEC_CFG_CMN = $CFG_CMN;
$DEC_CFG_VARNORM = $CFG_VARNORM;

$DEC_CFG_QMGR_DIR = "$CFG_BASE_DIR/qmanager";
$DEC_CFG_LOG_DIR = "$CFG_BASE_DIR/logdir";
$DEC_CFG_MODEL_DIR = "$CFG_MODEL_DIR";

$DEC_CFG_DICTIONARY = "$CFG_BASE_DIR/etc/el-gr.dic";
$DEC_CFG_FILLERDICT = "$CFG_BASE_DIR/etc/el-gr.filler";
$DEC_CFG_LISTOFFILES = "$CFG_BASE_DIR/etc/fot_test.fileids";
$DEC_CFG_TRANSCRIPTFILE = "$CFG_BASE_DIR/etc/fot_test.transcription";
$DEC_CFG_RESULT_DIR = "$CFG_BASE_DIR/result";
$DEC_CFG_PRERESULT_DIR = "$CFG_BASE_DIR/presult";

# This variables, used by the decoder, have to be user defined, and
# may affect the decoder output
```



```

$DEC_CFG_LANGUAGEMODEL = "$CFG_BASE_DIR/etc/el-gr.lm.bin";
# Or can be JSFG or FSG too, used if uncommented
# $DEC_CFG_GRAMMAR = "$CFG_BASE_DIR/etc/${CFG_DB_NAME}.jsfg";
# $DEC_CFG_FSG = "$CFG_BASE_DIR/etc/${CFG_DB_NAME}.fsg";

$DEC_CFG_LANGUAGEWEIGHT = "10";
$DEC_CFG_BEAMWIDTH = "1e-80";
$DEC_CFG_WORDBEAM = "1e-40";
$DEC_CFG_WORDPENALTY = "0.2";

$DEC_CFG_ALIGN = "builtin";

$DEC_CFG_NPART = 1;          # Define how many pieces to split decode in

# This variable has to be defined, otherwise utils.pl will not load.
$CFG_DONE = 1;

```

```
return 1;
```

Δημιουργία Ελληνικού μοντέλου για χρήση σε ROV

Φράσεις που χρησιμοποιήθηκαν για την εκπαίδευση του μοντέλου

ξεκίνα πορεία ευθεία σιγά
 στρίψε δεξιά στρίψε αριστερά σιγά σιγά
 σταμάτα ξεκίνα συνέχισε κάνε πίσω
 αύξησε ταχύτητα μείωσε ταχύτητα
 εκατόν ογδόντα μοίρες
 κάνε στροφή σαράντα πέντε μοιρών
 πάρε δείγμα πιάσε αντικείμενο
 άνοιξε κλείσε δίκτυο κλείσε δαγκάνα
 πορεία εκατόν είκοσι μοίρες
 πιο ήρεμα εξήντα μοίρες
 ακόμη λίγο ανέβασε ένταση
 μείωσε ένταση άνοιξε βραχίονα
 κλείσε βραχίονα δεξιό αριστερό
 εμπρός πίσω δεξιά αριστερά ευθεία
 κάνε κύκλο ενενήντα μοίρες άνοιξε φώτα
 άφησε αντικείμενο άνοιξε δάγκανα
 πορεία διακόσιες εβδομήντα μοίρες
 ξεκίνα ένα σταμάτα ένα
 ένα πιο πολύ ένα πιο λίγο
 ξεκίνα δύο σταμάτα δύο
 δύο πιο πολύ δύο πιο λίγο
 ξεκίνα τρία σταμάτα τρία
 τρία πιο πολύ τρία πιο λίγο
 ξεκίνα τέσσερα σταμάτα τέσσερα

τέσσερα πιο πολύ τέσσερα πιο λίγο
ξεκίνα πέντε σταμάτα πέντε
πέντε πιο πολύ πέντε πιο λίγο
τριακόσιες εξήντα μοίρες κλείσε φώτα
πάνω κάτω δεξιά αριστερά σιγά γρήγορα
δείγμα αντικείμενο άνοιξε κλείσε

Αποτελέσματα εκπαίδευσης Ελληνικού μοντέλου προσαρμοσμένο για χρήση σε εργασίες ROV

ξεκίνα πορεία ευθεία σιγά (user-487)

ξεκίνα πορεία ευθεία σιγά (user-487)

Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

στρίψε δεξιά αριστερά σιγά σιγά (user-488)

στρίψε δεξιά αριστερά σιγά σιγά (user-488)

Words: 5 Correct: 5 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

σταμάτα ξεκίνα συνέχισε κάνε πίσω (user-489)

σταμάτα ξεκίνα συνέχισε κάνε πίσω (user-489)

Words: 5 Correct: 5 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

αύξησε ταχύτητα μείωσε ταχύτητα (user-490)

αύξησε ταχύτητα μείωσε ταχύτητα (user-490)

Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

εκατόν ογδόντα μοίρες (user-491)

εκατόν ογδόντα μοίρες (user-491)

Words: 3 Correct: 3 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

κάνε στροφή σαράντα πέντε μοιρών (user-492)

κάνε στροφή σαράντα πέντε μοιρών (user-492)

Words: 5 Correct: 5 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

πάρε δείγμα πιάσε αντικείμενο (user-493)

πάρε δείγμα πιάσε αντικείμενο (user-493)

Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

άνοιξε κλείσε δίκτυο κλείσε δαγκάνα (user-494)

άνοιξε κλείσε δίκτυο κλείσε δαγκάνα (user-494)

Words: 5 Correct: 5 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

πορεία εκατόν είκοσι μοίρες (user-495)

πορεία εκατόν είκοσι μοίρες (user-495)

Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

πιο ήρεμα εξήντα μοίρες (user-496)

πιο ήρεμα εξήντα μοίρες (user-496)

Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

ακόμη λίγο ανέβασε ένταση (user-497)

ακόμη λίγο ανέβασε ένταση (user-497)

Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

μείωσε ένταση άνοιξε βραχίονα (user-498)

μείωσε ένταση άνοιξε βραχίονα (user-498)

Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

κλείσε βραχίονα δεξιό αριστερό (user-499)

κλείσε βραχίονα δεξιό αριστερό (user-499)

Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

εμπρός πίσω *** ** ΑΡΙΣΤΕΡΑ ΕΥΘΕΙΑ (user-500)

εμπρός πίσω ΑΠΟ ΤΑ ΣΤΕΝΑ ΚΑΙ ΘΑ (user-500)

Words: 4 Correct: 2 Errors: 5 Percent correct = 50.00% Error = 125.00% Accuracy = -25.00%

Insertions: 3 Deletions: 0 Substitutions: 2

κάνε κύκλο ενενήντα μοίρες άνοιξε φώτα (user-501)

κάνε κύκλο ενενήντα μοίρες άνοιξε φώτα (user-501)

Words: 6 Correct: 6 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

άφησε αντικείμενο άνοιξε ΔΑΓΚΑΝΑ (user-502)

άφησε αντικείμενο άνοιξε ΔΑΓΚΑΝΑ (user-502)

Words: 4 Correct: 3 Errors: 1 Percent correct = 75.00% Error = 25.00% Accuracy = 75.00%

Insertions: 0 Deletions: 0 Substitutions: 1

πορεία διακόσιες εβδομήντα μοίρες (user-503)

πορεία διακόσιες εβδομήντα μοίρες (user-503)

Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

ξεκίνα ένα σταμάτα ένα (user-504)

ξεκίνα ένα σταμάτα ένα (user-504)

Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

ένα πιο πολύ *** *** ένα πιο λίγο (user-505)

ένα πιο πολύ ΈΝΑ ΠΙΟ ένα πιο λίγο (user-505)

Words: 6 Correct: 6 Errors: 2 Percent correct = 100.00% Error = 33.33% Accuracy = 66.67%

Insertions: 2 Deletions: 0 Substitutions: 0

ξεκίνα δύο σταμάτα δύο (user-506)

ξεκίνα δύο σταμάτα δύο (user-506)

Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

δύο πιο πολύ δύο πιο λίγο (user-507)

δύο πιο πολύ δύο πιο λίγο (user-507)

Words: 6 Correct: 6 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

ξεκίνα τρία σταμάτα τρία (user-508)

ξεκίνα τρία σταμάτα τρία (user-508)

Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

τρία πιο πολύ τρία πιο λίγο (user-509)

τρία πιο πολύ τρία πιο λίγο (user-509)

Words: 6 Correct: 6 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

ξεκίνα τέσσερα σταμάτα τέσσερα (user-510)

ξεκίνα τέσσερα σταμάτα τέσσερα (user-510)

Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

τέσσερα πιο πολύ τέσσερα πιο λίγο (user-511)

τέσσερα πιο πολύ τέσσερα πιο λίγο (user-511)

Words: 6 Correct: 6 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

ξεκίνα πέντε σταμάτα πέντε (user-512)

ξεκίνα πέντε σταμάτα πέντε (user-512)

Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

πέντε πιο πολύ πέντε πιο λίγο (user-513)

πέντε πιο πολύ πέντε πιο λίγο (user-513)

Words: 6 Correct: 6 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

ΤΡΙΑΚΟΣΙΕΣ εξήντα μοίρες κλείσε φώτα (user-514)

ΤΡΟΠΟΣ εξήντα μοίρες κλείσε φώτα (user-514)

Words: 5 Correct: 4 Errors: 1 Percent correct = 80.00% Error = 20.00% Accuracy = 80.00%

Insertions: 0 Deletions: 0 Substitutions: 1

πάνω κάτω δεξιά αριστερά σιγά γρήγορα (user-515)

πάνω κάτω δεξιά αριστερά σιγά γρήγορα (user-515)

Words: 6 Correct: 6 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

δείγμα αντικείμενο άνοιξε κλείσε (user-516)

δείγμα αντικείμενο άνοιξε κλείσε (user-516)

Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

ξεκίνα πορεία ευθεία σιγά (user-611)

ξεκίνα πορεία ευθεία σιγά (user-611)

Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

στρίψε δεξιά στρίψε ΑΡΙΣΤΕΡΑ ΣΙΓΑ ΣΙΓΑ (user-612)

στρίψε δεξιά στρίψε *** ΣΥΜΒΑΣΕΩΝ ΕΣ'ΕΝΑ (user-612)

Words: 6 Correct: 3 Errors: 3 Percent correct = 50.00% Error = 50.00% Accuracy = 50.00%

Insertions: 0 Deletions: 1 Substitutions: 2

σταμάτα ξενίνα συνέχισε κάνε πίσω (user-613)

σταμάτα ξενίνα συνέχισε κάνε πίσω (user-613)

Words: 5 Correct: 5 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

*** ΑΥΞΗΣΕ ΤΑΧΥΤΗΤΑ ΜΕΙΩΣΕ ΤΑΧΥΤΗΤΑ (user-614)

ΟΠΩΣ ΕΠΟΧΗ ΤΑ ΑΝΘΗ ΣΤΟΧΟ (user-614)

Words: 4 Correct: 0 Errors: 5 Percent correct = 0.00% Error = 125.00% Accuracy = -25.00%

Insertions: 1 Deletions: 0 Substitutions: 4

εκατόν ογδόντα μοίρες (user-615)

εκατόν ογδόντα μοίρες (user-615)

Words: 3 Correct: 3 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

κάνε στροφή σαράντα πέντε μοιρών (user-616)

κάνε στροφή σαράντα πέντε μοιρών (user-616)

Words: 5 Correct: 5 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

πάρε δείγμα πιάσε αντικείμενο (user-617)

πάρε δείγμα πιάσε αντικείμενο (user-617)

Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

άνοιξε κλείσε δίκτυο κλείσε δαγκάνα (user-618)

άνοιξε κλείσε δίκτυο κλείσε δαγκάνα (user-618)

Words: 5 Correct: 5 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

πορεία εκατόν είκοσι μοίρες (user-619)

πορεία εκατόν είκοσι μοίρες (user-619)

Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

πιο ήρεμα εξήντα μοίρες (user-620)

πιο ήρεμα εξήντα μοίρες (user-620)

Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

ακόμη λίγο ανέβασε ένταση (user-621)

ακόμη λίγο ανέβασε ένταση (user-621)

Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

μείωσε ένταση άνοιξε βραχίονα (user-622)

μείωσε ένταση άνοιξε βραχίονα (user-622)

Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

κλείσε βραχίονα δεξιό αριστερό (user-623)

κλείσε βραχίονα δεξιό αριστερό (user-623)

Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

εμπρός πίσω δεξιά αριστερά ευθεία (user-624)

εμπρός πίσω δεξιά αριστερά ευθεία (user-624)

Words: 5 Correct: 5 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

κάνε κύκλο ενενήντα μοίρες άνοιξε φώτα (user-625)

κάνε κύκλο ενενήντα μοίρες άνοιξε φώτα (user-625)

Words: 6 Correct: 6 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

άφησε αντικείμενο άνοιξε ΔΑΓΚΑΝΑ (user-626)

άφησε αντικείμενο άνοιξε ΔΑΓΚΑΝΑ (user-626)

Words: 4 Correct: 3 Errors: 1 Percent correct = 75.00% Error = 25.00% Accuracy = 75.00%

Insertions: 0 Deletions: 0 Substitutions: 1

πορεία διακόσιες εβδομήντα μοίρες (user-627)

πορεία διακόσιες εβδομήντα μοίρες (user-627)

Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

ξεκίνα ένα σταμάτα ένα (user-628)

ξεκίνα ένα σταμάτα ένα (user-628)

Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

ένα πιο πολύ ένα πιο λίγο (user-629)

ένα πιο πολύ ένα πιο λίγο (user-629)

Words: 6 Correct: 6 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

ξεκίνα δύο σταμάτα δύο (user-630)

ξεκίνα δύο σταμάτα δύο (user-630)

Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

δύο πιο πολύ δύο πιο λίγο (user-631)

δύο πιο πολύ δύο πιο λίγο (user-631)

Words: 6 Correct: 6 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

ξεκίνα τρία σταμάτα τρία (user-632)

ξεκίνα τρία σταμάτα τρία (user-632)

Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

τρία πιο πολύ τρία πιο λίγο (user-633)

τρία πιο πολύ τρία πιο λίγο (user-633)

Words: 6 Correct: 6 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

ξεκίνα τέσσερα σταμάτα τέσσερα (user-634)

ξεκίνα τέσσερα σταμάτα τέσσερα (user-634)

Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

τέσσερα πιο πολύ τέσσερα πιο λίγο (user-635)

τέσσερα πιο πολύ τέσσερα πιο λίγο (user-635)

Words: 6 Correct: 6 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

ξεκίνα πέντε σταμάτα πέντε (user-636)

ξεκίνα πέντε σταμάτα πέντε (user-636)

Words: 4 Correct: 4 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

πέντε πιο πολύ πέντε πιο λίγο (user-637)

πέντε πιο πολύ πέντε πιο λίγο (user-637)

Words: 6 Correct: 6 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

τριακόσιες εξήντα μοίρες κλείσε φώτα (user-638)

τριακόσιες εξήντα μοίρες κλείσε φώτα (user-638)

Words: 5 Correct: 5 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

πάνω κάτω δεξιά αριστερά σιγά γρήγορα (user-639)

πάνω κάτω δεξιά αριστερά σιγά γρήγορα (user-639)

Words: 6 Correct: 6 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy = 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

δείγμα αντικείμενο άνοιξε ΚΛΕΙΣΕ (user-640)

δείγμα αντικείμενο άνοιξε ΤΗΣ (user-640)

Words: 4 Correct: 3 Errors: 1 Percent correct = 75.00% Error = 25.00% Accuracy = 75.00%

Insertions: 0 Deletions: 0 Substitutions: 1

TOTAL Words: 278 Correct: 265 Errors: 19

TOTAL Percent correct = 95.32% Error = 6.83% Accuracy = 93.17%

TOTAL Insertions: 6 Deletions: 1 Substitutions: 12

ΛΙΣΤΑ ΕΙΚΟΝΩΝ-ΣΧΕΔΙΑΓΡΑΜΜΑΤΩΝ

ΕΙΚΟΝΕΣ		
ΑΑ	ΠΕΡΙΓΡΑΦΗ	
1	Η ανάπτυξη του διαδικτύου μέσα από αριθμούς	Σελ. 3
2	Χρήση του CMU Sphinx για τις ανάγκες του Ευρ. Ερευνητικού προγράμματος RAPP	Σελ. 5
3	Η πλατφόρμα συγκριτικής αναζήτησης skrouz	Σελ. 10
4	Σχέδια από την σπηλιά του Chauvet	Σελ. 11
5	Πικτόγραμμα του 1510 που αναφέρεται στην έλευση μισθοφόρων στο νησί Ισπανιόλα	Σελ. 12
6	Αιγυπτιακά ιερογλυφικά	Σελ. 12
7	Κρητικά ιερογλυφικά πάνω σε πράσινη σφραγίδα Ίαση. 1800 π.Χ	Σελ. 13
8	Δέκτης σημάτων Morse	Σελ. 13
9	Υπολογιστής σε σπίτι το 1965	Σελ. 15
10	Προσωπικός υπολογιστής της δεκαετίας του 1980	Σελ. 16
11	Η αργική πρόταση για το www	Σελ. 16
12	Χρήστες υπηρεσιών Internet Μάρτιος 2017	Σελ. 17
13	Γραπτός και προφορικός λόγος	Σελ. 19
14	Έλεγχος συσκευών με την χρήση προφορικού λόγου	Σελ. 19
15	Καταγραφή προφορικού λόγου σε κινητή συσκευή	Σελ. 21
16	Ο λίθος της Ροζέττας με την επιγραφή του σε τρεις διαφορετικές Γλώσσες	Σελ. 22
17	Γραφική παράσταση που φέρει ο Pioneer	Σελ. 22
18	Η εφαρμογή της Apple-siri	Σελ. 26
19	Cockpit Eurofighter Typhoon	Σελ. 27
20	Το μαχητικό 4ης γενιάς JAS 39 Gripen	Σελ. 27
21	Κυματική μορφή προφορικού λόγου	Σελ. 38
22	Απλό διπλό και τριπλό φώνημα της Αγγλικής λέξης bat	Σελ. 38
23	Τυπική χρήση εργαλείων δημιουργίας Γλωσσικού μοντέλου	Σελ. 51
24	Παράδειγμα μετατροπής λέξεων με την χρήση του ARPABET	Σελ. 55
25	Δείγμα ηχητικού σήματος που χρησιμοποιήθηκε στην διαδικασία εκπαίδευσης	Σελ. 59
26	Το Ελληνικό μοντέλο στο sourceforge	Σελ. 64
27	Διαμόρφωση συστήματος Max Rover ROV	Σελ. 65
28	Τύποι καλωδίων που χρησιμοποιούνται στα ROV	Σελ. 66
29	Κονσόλες χειρισμού Max Rover ROV	Σελ. 66

30	Εργαλεία για βραχιόνες ROV	Σελ. 68
21	Χειριστήρια συστήματος βραχιόνων Hydrolek	Σελ. 69
32	Χειριστήρια συστήματος βραχιόνων Schilling	Σελ. 69
33	Εργασίες βραχιόνια Hydrolek σε βάθος 1000 m	Σελ. 70
34	Αποτελέσματα εκπαίδευσης Ελληνικού μοντέλου προσαρμοσμένο για γρήση σε ROV	Σελ. 71
35	Cockpit ελικοπτέρου NH-90	Σελ. 72
36	Συμμετογή Ελληνικών δυνάμεων στην πολυεθνική άσκηση Αργοναύτης 2017	Σελ. 75
37	Το σύνολο των διατιθέμενων μοντέλων της πλατφόρμας CMU Sphinx	Σελ. 77

ΣΧΕΔΙΑΓΡΑΜΜΑΤΑ		
ΑΑ	ΠΕΡΙΓΡΑΦΗ	
1	Η διαχρονική εξέλιξη στα μέσα επικοινωνίας	Σελ. 14
2	Τα μέρη ενός τυπικού συστήματος προφορικού λόγου	Σελ. 28
3	Αρχιτεκτονική ενός συστήματος ASR	Σελ. 29
4	Αρχιτεκτονική λειτουργίας CMU Sphinx	Σελ. 41
5	Αρχιτεκτονική λειτουργίας του front end μπλοκ	Σελ. 42
6	Αρχιτεκτονική σχεδιασμού αποκωδικοποιητή Pocketsphinx	Σελ. 44
7	Ροή δεδομένων κατά την φάση της ακουστικής μοντελοποίησης	Σελ. 45
8	Πλάνο εργασιών δημιουργίας Ελληνικού μοντέλου για το CMU Sphinx	Σελ. 49
9	Ροή εργασιών κατά την δημιουργία του Γλωσσικού μοντέλου	Σελ. 53
10	Τα μέρη του Ελληνικού μοντέλου για το CMU Sphinx	Σελ. 76