



**ΕΛΛΗΝΙΚΟ ΜΕΣΟΓΕΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ**

**ΣΧΟΛΗ ΜΟΥΣΙΚΗΣ &  
ΟΠΤΟΑΚΟΥΣΤΙΚΩΝ ΤΕΧΝΟΛΟΓΙΩΝ**

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΜΟΥΣΙΚΗΣ  
ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ ΑΚΟΥΣΤΙΚΗΣ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**«ΧΡΗΣΗ ΠΛΑΤΦΟΡΜΑΣ ΕΞΥΓΙΝΟΥ ΒΟΗΘΟΥ ΩΣ  
ΣΥΣΤΗΜΑ ΑΝΑΠΑΡΑΓΩΓΗΣ ΠΟΛΥΜΕΣΩΝ»**

φοιτητής (Α.Μ. 1369)  
ΝΙΚΟΛΑΟΣ Σ. ΖΗΣΟΠΟΥΛΟΣ

επιβλέπων καθηγητής  
Δρ. ΠΑΝΑΓΙΩΤΗΣ ΖΕΡΒΑΣ  
ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ

**Ακαδ. Έτος  
2019 - 2020**

## ΠΕΡΙΕΧΟΜΕΝΑ

	σελίδα
<b>ΕΙΣΑΓΩΓΙΚΟ ΣΗΜΕΙΩΜΑ</b>	<b>3</b>
<b>ΠΕΡΙΛΗΨΗ</b>	<b>4</b>
<b>ABSTRACT</b>	<b>5</b>
<b>ΚΕΦΑΛΑΙΟ ΠΡΩΤΟ</b>	
1. ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΤΕΧΝΟΛΟΓΙΑ ΟΜΙΛΙΑΣ	6
1.1. Τεχνολογία Ομιλίας, Αισθητήρες Ήχου & Ενεργοποιητές	6
1.2. Τεχνητή Νοημοσύνη, Έξυπνοι Βοηθοί & Τεχνολογία Ομιλίας	14
1.3. Αναγνώριση προτύπων για την τεχνολογία ομιλίας	25
<b>ΚΕΦΑΛΑΙΟ ΔΕΥΤΕΡΟ</b>	
2. ΜΙΚΡΟΥΪΠΟΛΟΓΙΣΤΕΣ ΚΑΙ EMBEDDED DEVICES	26
2.1. Raspberry Pi και άλλες επιλογές Μικροϋπολογιστών	26
2.2. Ανασκόπηση Modules ήχου και ομιλίας για IoT συσκευές	27
2.3. State of the art: Speech Recognition	31
2.4. State of the art: Speech Synthesis	32
2.5. State of the art: Chat bots	34
<b>ΚΕΦΑΛΑΙΟ ΤΡΙΤΟ</b>	
3. Ο ΕΞΥΠΝΟΣ ΒΟΗΘΟΣ JASPER	36
3.1. Διαδικασία Εγκατάστασης και επιλογών βιβλιοθηκών	37
<b>ΚΕΦΑΛΑΙΟ ΤΕΤΑΡΤΟ</b>	
4. ΣΥΜΠΕΡΑΣΜΑΤΑ - ΑΝΑΦΟΡΕΣ	66
4.1. Συμπεράσματα Εργασίας	66
4.2. Προτάσεις	67
<b>ΠΑΡΑΡΤΗΜΑ</b>	<b>68</b>
<b>ΠΗΓΕΣ</b>	<b>71</b>

## **ΕΙΣΑΓΩΓΙΚΟ ΣΗΜΕΙΩΜΑ**

Η παρούσα μελέτη πραγματοποιήθηκε στα πλαίσια εκπόνησης της πτυχιακής εργασίας μου προκειμένου στην ολοκλήρωση των σπουδών μου, στο Τμήμα Μηχανικών Μουσικής Τεχνολογίας και Ακουστικής, της Σχολής Μουσικής και Οπτοακουστικών Τεχνολογιών, του Ελληνικού Μεσογειακού Πανεπιστημίου.

Το υλικό που παρουσιάζεται στο παρών έργο, δεν έχει υποβληθεί για οποιοδήποτε άλλο πτυχίο ή ως εργαλείο για επαγγελματικούς λόγους και είναι αποτέλεσμα δικής μου ανεξάρτητης εργασίας.

Μετά την έρευνα που διεξήγαγα, τελειώνοντας και παραδίδοντας τη μελέτη μου, αισθάνομαι πιο πλούσιος σε γνώσεις σχετικά με το αντικείμενο του εκπονήματος και θα προσπαθήσω να σταθώ αντάξιος σε επαγγελματικό επίπεδο και σύμφωνα με τα υψηλά εφόδια που μου δίδαξε η Σχολή μου και το εκπαιδευτικό σύστημα μέχρι σήμερα.

Ως εκ τούτου, οφείλω να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Παναγιώτη Ζέρβα, για την εμπιστοσύνη που μου έδειξε, τις γνώσεις και την πολύτιμη εμπειρία του που μου μεταλαμπάδευσε κατά τη διάρκεια των σπουδών και ασφαλώς την καταλυτική συμβολή του στην εκπόνηση του πτυχιακού μου συγγράμματος.

Επίσης, θέλω να ευχαριστήσω όλους τους διδάσκοντες – καθηγητές μου, οι οποίοι με τις γνώσεις, την πείρα και την κατάρτισή τους, φρόντισαν να με προετοιμάσουν κατάλληλα για την αυξημένων απαιτήσεων μετέπειτα πορεία μου.

Τέλος, νιώθω την ανάγκη να ευχαριστήσω τους γονείς και την οικογένειά μου, οι οποίοι με περιέβαλλαν με αμέριστη ψυχική συμπαράσταση αλλά και ηθική και οικονομική υποστήριξη, σε όλη τη διάρκεια των απαιτητικών ακαδημαϊκών μου σπουδών. Με υποστήριξαν στις δύσκολες ώρες της μελέτης μου και θεωρώ πως η απλόχερη αγάπη και οι συμβουλές τους, μου παρείχαν και θα εξακολουθήσουν να μου παρέχουν για το μέλλον, τα εφόδια για να πραγματοποιήσω τους στόχους, τα όνειρα και τις φιλοδοξίες μου.

**ΕΥΧΑΡΙΣΤΩ**

## ΠΕΡΙΛΗΨΗ

Όταν γίνεται λόγος για «**IoT Project**» αυτό εννοιολογικά σημαίνει «**Internet Of Things Project**». Επί της ουσίας πρόκειται για ένα σχέδιο μέσα στο οποίο συνηθισμένα πράγματα που χρησιμοποιεί ο άνθρωπος στην καθημερινότητά του, όπως τα φώτα, οι πόρτες του σπιτιού του, οι ηλεκτρικές του συσκευές, συνδέονται σ' ένα δίκτυο υπολογιστή ή σ' ένα ενσωματωμένο σύστημα, στη συνέχεια στο internet και μετατρέπονται σε «**έξυπνα αντικείμενα**» ή «**έξυπνα εργαλεία**». Παραδείγματα ενός ενσωματωμένου συστήματος είναι οι μικροϋπολογιστές **Raspberry Pi** και **Arduino Uno**, οι οποίοι είναι και οι πιο διαδεδομένοι.

Ο στόχος της εργασίας είναι η επισκόπηση της τεχνητής νοημοσύνης στο πεδίο της τεχνολογίας ομιλίας, σε συνύπαρξη με τους έξυπνους βοηθούς. Πιο συγκεκριμένα, παρουσιάζονται διάφορες εμπορικές λύσεις οι οποίες επιτρέπουν την ύπαρξη και λειτουργία μιας καλής πρακτικής εφαρμογής έξυπνων βοηθών στο σπίτι ή το χώρο εργασίας.

Η εργασία ολοκληρώνεται με την ανάπτυξη της διαδικασίας εγκατάστασης του ανοικτού κώδικα **έξυπνου βοηθού Jasper** και σταδιακά παρουσιάζεται η πρακτική που ακολουθείται ούτως ώστε να ενεργοποιηθεί.

Οι **λέξεις κλειδιά** που κυρίως χρησιμοποιήθηκαν στη μελέτη ήταν:

Internet Of Things, Τεχνολογία Ομιλίας, Μικροϋπολογιστές, Έξυπνοι Βοηθοί

## ABSTRACT

When it comes to an “**IoT project**” literally means the **Internet of Things Project**.

It is essentially a plan in which the ordinary things that a person uses in his daily life, such as the lights or the doors of his home, his electrical devices, are connected to a network computer or an embedded system, then on the internet and turn into “**smart objects**” or “**smart tools**”. Examples of an embedded system is the **Raspberry Pi** and **Arduino Uno** micro-computers, which are the most frequently used by the public.

The aim of this work is to review artificial intelligence in the field of speech technology in co-existence with intelligent assistants. In particular, various commercial solutions are presented that allow the existence and operation of a good actual assistance of an Intelligent Agent in the home or workplace.

The task is completed with the development of the open-source **intelligent agent “Jasper”** installation process and gradually presents the procedure which is mandatory to be followed in order to enable it.

The **keywords** that were mainly used in the study were:

Internet Of Things, Speech Technology, Microcomputers, Intelligent Agents

## ΚΕΦΑΛΑΙΟ ΠΡΩΤΟ

### 1. ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΤΕΧΝΟΛΟΓΙΑ ΟΜΙΛΙΑΣ

#### 1.1. Τεχνολογία Ομιλίας, Αισθητήρες Ήχου & Ενεργοποιητές

Η αναγνώριση φωνής ή ομιλίας είναι η τεχνολογία με την οποία οι ήχοι, οι λέξεις ή οι φράσεις που ομιλούνται από τον άνθρωπο μετατρέπονται σε ηλεκτρικά σήματα, και αυτά τα σήματα μετασχηματίζονται σε πρότυπα κωδικοποίησης στα οποία έχει αποδοθεί η έννοια αυτή. Η εν λόγω τεχνολογία χρειάζεται ένα συνδυασμό της βελτιωμένης τεχνογνωσίας της τεχνητής νοημοσύνης και μιας εξελιγμένης μηχανής αναγνώρισης ομιλίας.

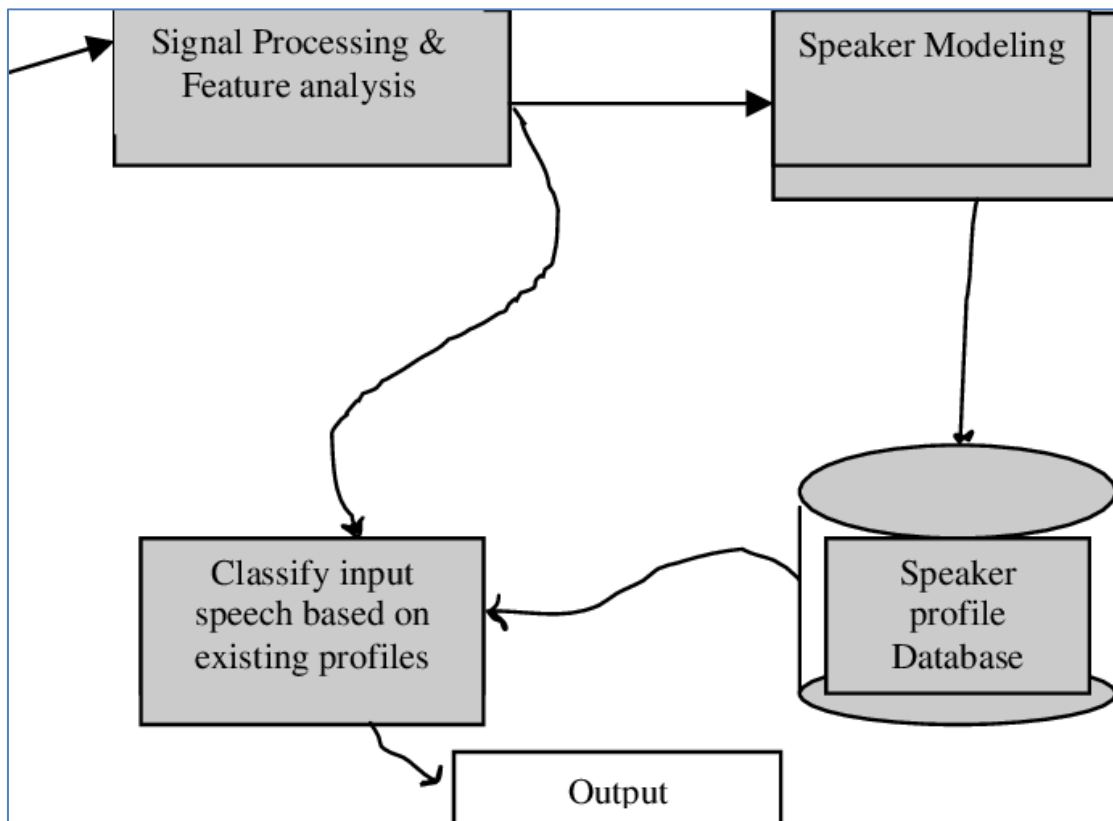
Η τεχνολογία αναγνώρισης ομιλίας παρέχει τρόπους με τους οποίους η τεχνολογία των υπολογιστών μπορεί να χρησιμοποιηθεί για εργασίες που χρησιμοποιούν φυσική γλώσσα, ή ομιλία. Τώρα έχουμε εισέλθει στην εποχή της επιστήμης των υπολογιστών που ασχολείται με την αλληλεπίδραση μεταξύ υπολογιστών και ανθρώπινης ομιλίας. Η αναγνώριση φωνής ή η αναγνώριση ομιλίας, είναι η τεχνολογία που χρησιμοποιεί είσοδο ήχου για την εισαγωγή δεδομένων και όχι πληκτρολόγιο. Μιλώντας σε ένα μικρόφωνο και παράγοντας το ίδιο αποτέλεσμα με την πληκτρολόγηση λέξεων με το χέρι μέσω πληκτρολόγιου. Ονομάζεται λογισμικό αναγνώρισης φωνής και έχει σχεδιαστεί με βάση δεδομένων με αναγνωρίσιμες λέξεις ή φράσεις. Το πρόγραμμα ταιριάζει με τις ηχητικές συχνότητες της ομιλίας με αντίστοιχες καταχωρήσεις στη βάση δεδομένων που του έχουν οριστεί.

### ***Ιστορική Αναδρομή***

Η μελέτη της αναγνώρισης ομιλίας και της μεταγραφής άρχισε το 1936 με τα Bell Labs της AT & T. Παλαιότερα, η έρευνα χρηματοδοτήθηκε και εκτελέστηκε από τα πανεπιστήμια και την κυβέρνηση των ΗΠΑ (κυρίως από το Στρατό και την DARPA - Προηγμένη Υπηρεσία Έρευνας Έργων Άμυνας). Στις αρχές της δεκαετίας του 1980, η τεχνολογία αναγνώρισης ομιλίας έφθασε στην εμπορική αγορά. Υπήρχαν πολλά ανταγωνιστικά ερευνητικά "στρατόπεδα", οργανωμένα το καθένα να εργάζεται ανεξάρτητα, για την ανάπτυξη της αναγνώρισης ομιλίας. Η πρώτη εταιρεία που ξεκίνησε ένα εμπορικό προϊόν ήταν η Convox το 1982. Η Convox έφερε ψηφιακό ήχο (μέσω του Voice Master, Sound Master και The Speech Thing) στο Commodore 64, Atari 400/800 και τελικά στο IBM PC στα μέσα του '80. Με αυτή την αλληλεπίδραση του ήχου με τους υπολογιστές, η ομιλία αναγνωρίστηκε σαν ύπαρξη. Μια άλλη εταιρεία που ιδρύθηκε το 1982 και της οποίας το προϊόν έχει γίνει ο διάσημος ηγέτης στην ομιλία αγοράς αναγνώρισης, ήταν η Dragon Systems Inc. (σήμερα Scan soft Inc). Τώρα κατέχει και κατασκευάζει αυτό το προϊόν, το Dragon Naturally Speaking.

Πώς γίνεται η αναγνώριση ομιλίας; Πώς ένας υπολογιστής μετατρέπει προφορικές λέξεις ή ομιλία στα δεδομένα που μπορεί να καταλάβει και στη συνέχεια να τα εκτελέσει αναλόγως; Ενώ χρησιμοποιείται η τεχνολογία αναγνώρισης ομιλίας πρέπει κάποιος να μιλήσει σ' ένα μικρόφωνο. Το ηλεκτρικό σήμα από το μικρόφωνο ψηφιοποιείται από αναλογικό σε ψηφιακό (A / D) και αποθηκεύεται στη μνήμη. Για να προσδιοριστεί η έννοια της φωνητικής εισόδου, ο υπολογιστής επιχειρεί να ταιριάξει την είσοδο με κάποιο ψηφιοποιημένο δείγμα φωνής ή πρότυπο που του έχει οριστεί. Κατά τη διάρκεια της αναγνώρισης ομιλίας, μια ροή δειγμάτων ήχου (audio samples) εισάγονται ως δεδομένα εισόδου στο πρόγραμμα. Προηγουμένως, θα πρέπει να είναι έτοιμη η βάση δεδομένων ούτως ώστε να ξεκινήσει η διαδικασία σύγκρισης χαρακτηριστικών εισόδου – ορισμάτων. Στη συνέχεια, τα χαρακτηριστικά αντιστοιχούν στο περιεχόμενο της βάσης δεδομένων. Αυτή η τεχνική είναι παρόμοια με τις παραδοσιακές εντολές εισόδου από ένα πληκτρολόγιο. Το πρόγραμμα περιέχει τα πρότυπα εισόδου και επιχειρεί

να τα ταιριάζει με αυτό που λαμβάνει σαν είσοδο. Πράγματι, στις αρχές της δεκαετίας του 1990, οι καλύτεροι αναγνωριστές ομιλίας, είχαν ένα ποσοστό σφάλματος 15% σε μια σχετικά μικρή εντολή υπαγόρευσης 20.000 λέξεων. Τώρα όμως, αυτό το ποσοστό σφάλματος έχει μειωθεί σε μόλις 1-2%, αν και αυτό δεν είναι απόλυτο. Ακολούθως φαίνεται το Διάγραμμα Ροής του λογισμικού Τεχνολογίας Ομιλίας.



Διάγραμμα 1.1. Μοντέλο Λειτουργίας του Λογισμικού Τεχνολογίας Ομιλίας

Οι εφαρμογές αναγνώρισης ομιλίας είναι διαφορετικές από οποιοδήποτε άλλο είδος εφαρμογής ηλεκτρονικού υπολογιστή. Ανοίγει έναν κόσμο γεμάτο δυνατότητες για τους προγραμματιστές. Οι εφαρμογές ομιλίας μπορούν επίσης να παρουσιάσουν στους χρήστες περισσότερες επιλογές ανά πάσα στιγμή, καθώς αυτές δεν περιορίζονται από τον αριθμό των πλήκτρων σε ένα πληκτρολόγιο του τηλεφώνου, ούτε οι χρήστες πρέπει να θυμούνται πολύπλοκες αριθμητικές τιμές. Οι χρήστες μπορούν απλά να πουν αυτό που θέλουν. Υπάρχουν τομείς όπου η αναγνώριση ομιλίας μπορεί να λειτουργήσει πολύ αποτελεσματικά.



### **ΤΟΜΕΙΣ ΑΝΑΓΝΩΡΙΣΗΣ ΟΜΙΛΙΑΣ**

#### **1. Άτομα με ειδικές ανάγκες ή αναπηρία**

Οι περιοχές που εκτιμάται περισσότερο η εφαρμογή της αναγνώρισης ομιλίας, είναι σε ανθρώπους με αναπηρία ή με ειδικές ανάγκες. Κατά την τελευταία δεκαετία, πολλές βελτιώσεις έχουν πραγματοποιηθεί στην εκτέλεση των αναγνωριστών ομιλίας και στη σημερινή τεχνολογία σε σχέση με τις ανάγκες των ατόμων αυτών. Οι τυφλοί ήταν οι πρώτοι οι οποίοι επωφελήθηκαν σημαντικά από την αναγνώριση ομιλίας τεχνολογίας υπολογιστών. Με τη χρήση απλών μηχανημάτων που θα μπορούσαν να διαβάσουν τις λέξεις στις πρώιμες οθόνες υπολογιστών που βασίζονται σε κείμενο και να τις μετατρέψουν σε συνθετική ομιλία, καθώς και να αποδεχθούν εντολές που ομιλούνται από κάποιον τυφλό. Αυτό, τους κατέστησε ικανούς να λειτουργούν αποτελεσματικά τους υπολογιστές. Επίσης, είναι χρήσιμο για τους ανθρώπους οι οποίοι έχουν κινητική δυσκολία ή δεν μπορούν να χρησιμοποιήσουν τα χέρια τους.

#### **2. Φροντίδα υγείας**

Η αναγνώριση ομιλίας είναι μια ευγενής τεχνολογία για τις σύγχρονες φροντίδες υγείας. Η πρόοδος της αναγνώρισης ομιλίας από τη δημιουργία κειμένου για την καταγραφή και την ανάγνωση των πληροφοριών είναι πολύ χρήσιμη για τους γιατρούς. Διευκολύνει τη συλλογή πληροφοριών στα ηλεκτρονικά συστήματα αρχείων υγείας και επιτρέπει στους γιατρούς να αλληλεπιδρούν με το σύστημα απλώς μέσω ομιλίας. Βέβαια, βρίσκεται ακόμη σε εξέλιξη, όμως γίνεται χρήση της σε ορισμένα τμήματα της υγειονομικής περίθαλψης, όπως π.χ. ραδιολογία, παθολογία και ιατρική έκτακτης ανάγκης. Ένα παράδειγμα συστήματος αναγνώρισης ομιλίας που είναι διαθέσιμο στην περιοχή της υγειονομικής περίθαλψης είναι η Dragon Natural Speaking Medical Solutions. Με την πρόοδο της τεχνολογίας εκτιμάται πως θα ενσωματωθούν μελλοντικές εφαρμογές, λύσεις περιεχομένου, όπως επίσης και τεχνολογία ιατρικών διαδρομών και κωδικοποίηση βάσεων δεδομένων.

### 3. Τηλεφωνία

Πρόσφατες εξελίξεις στην ασύρματη επικοινωνία και την αναγνώριση ομιλίας έχουν επιτρέψει την πρόσβαση στο διαδίκτυο από οπουδήποτε, ανά πάσα στιγμή, χρησιμοποιώντας μόνο ένα τηλέφωνο. Μερικές πιθανές εφαρμογές είναι η περιήγηση στον ιστό, επαλήθευση χρονοδιαγραμμάτων πτήσης, λήψη χαρτών και οδηγιών, έλεγχος ηλεκτρονικού ταχυδρομείου κλπ. Τα τηλέφωνα μπορούν επίσης να προσφέρουν φωνητική πρόσβαση στο διαδίκτυο. Τα πλεονεκτήματα πρόσβασης τηλεφώνου μέσω φωνής στον ιστό είναι αξιολογώσιμα:

- καταρχήν είναι εύκολο στη χρήση. Σε αντίθεση με τη διεπαφή υπολογιστή, μια φωνή δε χρειάζεται πληκτρολόγιο, κανένα ποντίκι, καμία οθόνη, απελευθερώνοντας τους χρήστες από αυτά τα εμπόδια στην πρόσβαση και τη δράση,
- είναι προσβάσιμο σε οποιονδήποτε με τη χρήση ενός έξυπνου τηλεφώνου,
- σημαντικό πλεονέκτημα είναι η πρόσβαση από οπουδήποτε. Η φωνή είναι πληροφορία η οποία μπορεί να αποσταλεί και να ανακτηθεί από οπουδήποτε. Οι χρήστες μπορούν να έχουν πρόσβαση οποιαδήποτε στιγμή και από οπουδήποτε.

Η φωνή καθιστά δυνατή την αποτελεσματικότερη χρήση του χρόνου. Η τεχνολογία αναγνώρισης ομιλίας έχει εκμεταλλευτεί από πολλές βιομηχανίες. Άτομα με ειδικές ανάγκες και δεξιότητες, είναι σπουδαίοι χρήστες αυτής της τεχνολογίας. Οι άνθρωποι που χρησιμοποιούν αναγνώριση ομιλίας λόγω αναπηρίας, συχνά έχουν επιπλέον κίνητρο να βοηθήσουν στη βελτίωση της τεχνολογίας. Επιπλέον, πωλητές λογισμικού τρίτων κατασκευαστών, παίρνουν τις δυνατότητες αναγνώρισης ομιλίας των υπολογιστών σε νέους τομείς, συμπεριλαμβανομένων του οικιακού αυτοματισμού και της κινητής τηλεφωνίας. Η τεχνολογία αναγνώρισης ομιλίας εξακολουθεί να βρίσκεται σε αρχικό στάδιο βέβαια. Σήμερα υπάρχουν τα βασικά συστήματα αναγνώρισης φωνής. Αυτό σημαίνει πως η τεχνολογία είναι πολύ νέα και έχει περιορισμένες δυνατότητες. Φυσικά, υπάρχουν προκλήσεις για αυτή την τεχνολογία. Πιθανός περιορισμός μπορεί να είναι το κόστος. Για να καταστεί η τεχνολογία αυτή προσβάσιμη στο ευρύ

κοινό, θα πρέπει να είναι οικονομικά προσιτή, στο μέσο όρο χρηστών. Με την έλευση των έξυπνων μικροφώνων φιλτραρίσματος και την σχεδόν τελειοποίηση της αναγνώριση ομιλίας μεταγενέστερα, αναμένεται ελπιδοφόρα μια νέα εποχή εξέλιξης και αλληλεπίδρασης ανθρώπου-υπολογιστή.

Η τεχνολογία ομιλίας<sup>1</sup> είναι μια τεχνολογία η οποία χτίζεται μέσω έρευνας αναγνώρισης ομιλίας. Ένα μηχάνημα, όπως ένας υπολογιστής, μπορεί να καταλάβει τι λέει ο άνθρωπος και να αναγνωρίσει κάθε μοντέλο ομιλίας, οπότε μπορεί να διακρίνει την ομιλία του καθενός. Η τεχνολογία ομιλίας καλύπτει θέματα όπως ομιλία-σε-κείμενο αλλά και κείμενο-σε-ομιλία. Κάποιοι ερευνητές έχουν ήδη προσδιορίσει αρκετά μοντέλα ομιλίας για κάποιες γλώσσες όπως για παράδειγμα Αγγλικά, Γερμανικά, Κινέζικα και Γαλλικά.

Ηχητικές πηγές μπορούν να παραχθούν από ανθρώπους, ζώα, αυτοκίνητα και άλλα. Για να επεξεργαστούν τα ηχητικά στοιχεία, πρέπει να μετατραπεί η ηχητική πηγή από φυσική σε ψηφιακή. Αυτό μπορεί να επιτευχθεί με τη χρήση συσκευών με αισθητήρες που μπορούν να λάβουν μια φυσική ηχητική πηγή. Ένας απλός αισθητήρας ήχου, είναι ένα μικρόφωνο. Ο αισθητήρας μπορεί να ηχογραφήσει οποιαδήποτε πηγή, μέσω του μικροφώνου. Επιπλέον, υπάρχει η δυνατότητα παραγωγής ήχου με τη χρήση ενός ενεργοποιητή. Ένας απλός ηχητικός ενεργοποιητής είναι ένας παθητικός βομβητής. Αυτό το εξάρτημα μπορεί να παράξει απλούς ήχους με περιορισμένη συχνότητα. Ο βομβητής είναι συνήθως παθητικός ενεργοποιητής. Αν θέλει κανείς να δουλέψει με ενεργητικό ηχητικό ενεργοποιητή, μπορεί να χρησιμοποιήσει ένα ηχείο.

Η φωνή είναι η πιο βασική, κοινή και αποτελεσματική μορφή επικοινωνιακής μεθόδου, που επιτρέπει στους ανθρώπους να αλληλεπιδρούν μεταξύ τους. Οι σημερινές τεχνολογίες συνήθως είναι διαθέσιμες για περιορισμένο αλλά ενδιαφέρον πεδίο εργασιών. Αυτές οι τεχνολογίες επιτρέπουν στις μηχανές να ανταποκρίνονται σωστά και αξιόπιστα στις ανθρώπινες φωνές και να παρέχουν χρήσιμες και πολύτιμες υπηρεσίες. Η επικοινωνία μεταξύ των ανθρώπων

---

<sup>1</sup> [https://en.wikipedia.org/wiki/Speech\\_recognition](https://en.wikipedia.org/wiki/Speech_recognition)

κυριαρχείται από την προφορική μορφή γλώσσας, επομένως είναι φυσικό οι άνθρωποι να περιμένουν φωνητικές διεπαφές με τον υπολογιστή. Αυτό μπορεί να επιτευχθεί με την ανάπτυξη συστήματος φωνητικής αναγνώρισης: ομιλία σε κείμενο, που αφορά στη διαδικασία μετατροπής ενός ηχητικού σήματος, που συλλαμβάνεται με τη χρήση ενός μικροφώνου, σε ένα σύνολο λέξεων κι έτσι ο υπολογιστής, να μεταφράζει τη φωνή και την υπαγορεύει σε κείμενο.

### **ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ ΑΝΑΓΝΩΡΙΣΗΣ ΟΜΙΛΙΑΣ**

Το σύστημα αναγνώρισης ομιλίας μπορεί να ταξινομηθεί σε διάφορους τύπους, περιγράφοντας τον τύπο της ομιλίας, τον τύπο του μοντέλου ομιλητή και τον τύπο της ικανότητας που αναγνωρίζουν. Παρακάτω θα αναλυθούν οι προκλήσεις που προκύπτουν.

Τύποι Ομιλίας: Η αναγνώριση ομιλίας ταξινομείται ανάλογα με το είδος της έκφρασης που δύναται να αναγνωρισθεί, ως ακολούθως:

- 1) Απομονωμένη λέξη: ο απομονωμένος αναγνωριστής λέξεων συνήθως απαιτεί κάθε ομιλούμενη λέξη να έχει ησυχία, δηλαδή έλλειψη ήχου στο παρασκήνιο. Αποδέχεται μία λέξη κάθε φορά.
- 2) Συνδεδεμένη λέξη: είναι παρόμοια με την απομονωμένη λέξη, αλλά επιτρέπει ξεχωριστές δηλώσεις που περιέχουν μια ελάχιστη παύση μεταξύ τους.
- 3) Συνεχής ομιλία: επιτρέπει στους χρήστες να μιλούν φυσικά και παράλληλα ο υπολογιστής να καθορίσει το περιεχόμενο.
- 4) Αυθόρμητος Λόγος: είναι ο τύπος ομιλίας που είναι φυσικός ήχος.

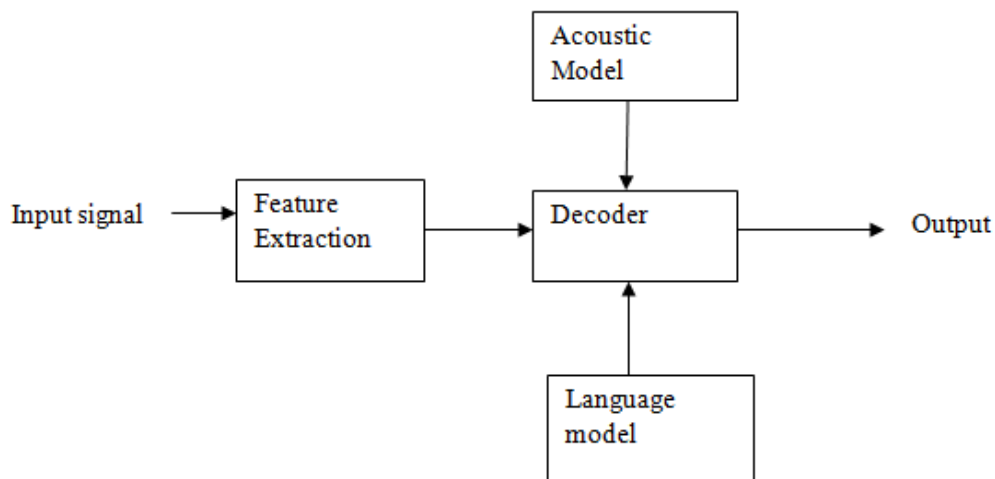
Το σύστημα αναγνώρισης ομιλίας βασίζεται σε χωριστές κατηγορίες που ορίζονται από τα μοντέλα ομιλητών και συγκεκριμένα, εξαρτώμενος ομιλητής και ανεξάρτητος ομιλητής. Αναλυτικά:

- 1) Μοντέλα εξαρτώμενα από ηχεία: αυτά τα συστήματα έχουν σχεδιαστεί για ένα συγκεκριμένο ηχείο. Είναι ευκολότερο να αναπτυχθούν και αποδίδουν μεγαλύτερη ακρίβεια, αλλά δεν είναι τόσο ευέλικτα.
- 2) Μοντέλα ανεξάρτητα από ηχεία: τα συστήματα αυτά είναι σχεδιασμένα για ποικιλία ηχείων. Είναι δυσκολότερο να αναπτυχθούν και εμφανίζονται λιγότερο ακριβή, αλλά είναι πολύ πιο ευέλικτα.

3) Τύποι λεξιλογίου: το μέγεθος του λεξιλογίου του συστήματος αναγνώρισης ομιλίας επηρεάζει τις απαιτήσεις επεξεργασίας, την ακρίβεια και την πολυπλοκότητα του συστήματος. Στο σύστημα αναγνώρισης φωνής ομιλίας σε κείμενο, οι τύποι λεξιλογίου μπορούν να ταξινομηθούν ως εξής:

- I. Μικρό λεξιλόγιο: ένα γράμμα.
- II. Μεσαίο λεξιλόγιο: δύο ή τρία γράμματα.
- III. Μεγάλο λεξιλόγιο: περισσότερες λέξεις γραμμάτων.

Παρακάτω φαίνεται μια επισκόπηση του συστήματος αναγνώρισης φωνής κειμένου σε ομιλία:



*Διάγραμμα 1.2. Λογικό Διάγραμμα Συστήματος Αναγνώρισης Φωνής: κείμενο σε ομιλία*

Αναλύοντας αυτό που φαίνεται στο παραπάνω διάγραμμα, προκύπτουν τα εξής συμπεράσματα:

Αρχικά, υπάρχει η είσοδος του σήματος, η είσοδος φωνής από το χρήστη. Συνεχίζοντας, πραγματοποιείται η διαδικασία της εξαγωγής, κατά την οποία θα πρέπει να διατηρηθούν χρήσιμες πληροφορίες για το σήμα, να αφαιρεθούν περιττές και ανεπιθύμητες πληροφορίες, να παρουσιάζονται λιγότερες διακυμάνσεις από το ένα περιβάλλον ομιλίας σε άλλο και να εμφανίζονται κανονικά και φυσικά μέσα ομιλίας. Σειρά έχει το Ακουστικό Μοντέλο, το οποίο περιέχει στατιστικές παραστάσεις κάθε ξεχωριστού ήχου που συνθέτει μια λέξη. Στη συνέχεια ακολουθεί η αποκωδικοποίηση. Θα αποκωδικοποιηθεί το σήμα εισόδου μετά την εξαγωγή χαρακτηριστικών και θα δείξει την επιθυμητή έξοδο.

Τέλος, υπάρχει το γλωσσικό μοντέλο, του οποίου ευθύνη είναι να αποδίδει μια πιθανότητα σε μια ακολουθία λέξεων μέσω μιας κατανομής πιθανοτήτων. Το κείμενο ερμηνείας εξόδου δίνεται από τον υπολογιστή.

### **1.2. Τεχνητή Νοημοσύνη, Έξυπνοι Βοηθοί & Τεχνολογία Ομιλίας**

Οι Έξυπνοι Προσωπικοί Βοηθοί (Ε.Π.Β.) έχουν εφαρμογή και χρησιμοποιούνται σε Λειτουργικά Συστήματα, στο Δίκτυο των Πραγμάτων και σε πληθώρα άλλων συστημάτων. Σήμερα, υπάρχουν πολλές εφαρμογές των έξυπνων βοηθών και εταιρίες, όπως η Apple, η Google και η Microsoft, τις χρησιμοποιούν σαν ένα μεγάλο χαρακτηριστικό στα λειτουργικά τους συστήματα και τις συσκευές τους. Με την Επεξεργασία Φυσικής Γλώσσας (Natural Language Processing), τη Μηχανική Μάθηση (Machine Learning), την Τεχνητή Νοημοσύνη (Artificial Intelligence) και μοντέλα πρόβλεψης, από αυτά τα πεδία, στον τομέα της Επιστήμης Υπολογιστών (Computer Science), αλλά και θεωρίες και τεχνικές μέσω Αλληλεπίδρασης Ανθρώπου – Υπολογιστή (Human – Computer Interaction), οι Ε.Π.Β έχουν γίνει πιο έξυπνοι και σχετικοί από ποτέ.

Οι Ε.Π.Β. και η τρέχουσα λειτουργική εφαρμογή τους εμφανίζονται σ' ένα εύρος εφαρμογών, συνήθως ενσωματωμένο σε λειτουργικά συστήματα, από διάφορους προγραμματιστές και οργανισμούς, όπως για παράδειγμα στους Ηλεκτρονικούς Υπολογιστές, στα κινητά τηλέφωνα και στο Δίκτυο των Πραγμάτων. Οι Ε.Π.Β. είναι ενσωματωμένοι σε διαφορετικές προγραμματιστικές γλώσσες και συμπεριφέρονται ποικίλα, αλλά όλοι εμπίπτουν στους ίδιους κλάδους της Επιστήμης των Υπολογιστών και παρουσιάζουν παρεμφερείς δυσκολίες. Η Επεξεργασία Φυσικής Γλώσσας, και η Αλληλεπίδραση Ανθρώπου-Υπολογιστή πάντα ήταν ουσιώδη σημεία γι' αυτό τον τύπο συστημάτων και, πιο πρόσφατα, καθώς τα συστήματα εξελίσσονται περαιτέρω, η Μηχανική Μάθηση και η Τεχνητή Νοημοσύνη έχουν γίνει αναπόσπαστα κομμάτια για τη λειτουργικότητα των Ε.Π.Β. Apple, Google και Microsoft είναι οργανισμοί οι οποίοι κατέχουν τις δικές τους μορφές τέτοιου είδους συστημάτων και συνεχόμενα τις ενσωματώνουν σε όλη την έκταση των εφαρμογών, συσκευών και συστημάτων τους.

### Πεδίο Εφαρμογής και Στόχοι

Στην παρούσα μελέτη, παρουσιάζεται θεωρητικά η έννοια του κλάδου της Επιστήμης Υπολογιστών, η οποία είναι θεμελιώδης για να οικοδομηθεί ένας Ε.Π.Β. Ακολουθεί αναφορά στις υπάρχουσες εφαρμογές επιχειρήσεων, από 3 διαφορετικές εταιρίες (Apple, Google, Microsoft), με σκοπό την ανάλυση και σύγκριση των συμπεριφορών τους.

### Τεχνητή Νοημοσύνη (T.N.)

Η T.N. είναι ένα επιστημονικό πεδίο το οποίο στοχεύει στο να κατανοήσει και να χτίσει τις έξυπνες οντότητες. Οι υπάρχοντες επίσημοι ορισμοί της T.N. σχετίζονται με διάφορες παραμέτρους όπως η συμπεριφορά, η επεξεργασία σκέψης και η λογική. Η διάκριση μεταξύ ανθρώπου και ορθολογικής συμπεριφοράς συχνά αναφέρεται στο πεδίο της. Για να δημιουργηθεί T.N. τα δύο συστατικά της *ευφυΐας* και των *εργαλείων* είναι απαραίτητα. Το πεδίο της Επιστήμης Υπολογιστών έχει δημιουργήσει τέτοια εργαλεία.

### Βοηθοί

Ένας Βοηθός (Agent) είναι μια οντότητα η οποία παρουσιάζει δράση. Τα υπολογιστικά προγράμματα πάντοτε δρουν και εκτελούν κάτι, ωστόσο από έναν Βοηθό περιμένουμε κάτι παραπάνω. Ένας Βοηθός αναμένεται να είναι σε θέση να δράσει αυτόνομα, να αναλύσει, να προσαρμοστεί στο περιβάλλον του και να αλλάξει, να επιμένει μέσα σε μεγάλο χρονικό διάστημα και τελικά να καταφέρνει να δημιουργεί, να αντιλαμβάνεται και να εκτελεί στόχους.

### Ορθολογικός Βοηθός

Ένας Ορθολογικός Βοηθός είναι ένας βοηθός ο οποίος για κάθε δράση και στόχο, λειτουργεί με τέτοιο τρόπο ώστε να στοχεύει στο καλύτερο δυνατό αποτέλεσμα. Όταν υπάρχουν αβεβαιότητες ο Ορθολογικός Βοηθός θεωρείται ότι θα επιτύχει το καλύτερο δυνατό αποτέλεσμα.

### Έξυπνος Βοηθός

Ένας Έξυπνος Βοηθός είναι ένας τύπος βοηθού που θεωρείται ορθολογικός. Με τη χρήση ενός συνόλου αρχών σχεδιασμού, οι προγραμματιστές είναι σε θέση να δημιουργήσουν επιτυχώς βοηθούς, οι οποίοι κάνουν το σύστημα λογικά ευφύες, ώστε να επιτευχθούν συγκεκριμένοι έξυπνοι στόχοι. Οι Ε.Π.Β. είναι λογισμικό που λειτουργεί εκ μέρους του χρήστη έτσι, ώστε να ολοκληρωθούν στόχοι και να παρέχουν πληροφορίες. Η επικοινωνία μεταξύ οντότητας και χρήστη συνήθως βασίζεται σε φωνητικές εντολές.

### Στατιστική Μάθηση

Περιλαμβάνει ορισμούς συγκεκριμένων πεδίων τα οποία είναι σχετικά με τη δημιουργία των Ε.Π.Β. Η Τεχνητή Νοημοσύνη (Τ.Ν.), η Μηχανική Μάθηση (Μ.Μ.) και η ανάλυση συναισθημάτων εισάγονται μαζί με το γενικότερο πλαίσιο τους και τη σχετικότητά με τους Ε.Π.Β.

### Μηχανική Μάθηση (Μ.Μ.)

Η Μ.Μ. είναι υποπεδίο της Τ.Ν. το οποίο στοχεύει στην εφαρμογή αλγορίθμων που μπορούν να μάθουν αυτόνομα. Στατιστικές και μαθηματικές βελτιστοποιήσεις παρέχουν μεθόδους και εφαρμογές στο πεδίο της Μ.Μ. λόγω του ισχυρού δεσμού τους. Και τα δύο πεδία στοχεύουν στον εντοπισμό ενδιαφέροντος προτύπου από τα δεδομένα. Ένα μεγάλο ζήτημα αλλά και μειονέκτημα για τη χρήση της Μ.Μ. και την κατηγοριοποίηση μοντέλων είναι το ρίσκο της υπέρ καταλληλότητας, το οποίο συμβαίνει όταν ένας αλγόριθμος υπερεκτιμά τις παραμέτρους στα δεδομένα εκμάθησης. Το αντίθετο από αυτό είναι το λεγόμενο bootstrapping, το οποίο σημαίνει να δημιουργηθούν κατασκευασμένα δεδομένα με στατιστικά μοντέλα με τη βοήθεια ενός μικρού δείγματος δεδομένων.



### Φυσική Επεξεργασία Γλώσσας (Φ.Ε.Γ.)

Η Φ.Ε.Γ. είναι ένα πεδίο στην Τ.Ν. και γλωσσολογία που αφορά την αλληλεπίδραση μεταξύ υπολογιστών και ανθρώπινης φυσικής γλώσσας. Ως μέρος και συστατικό της Αλληλεπίδρασης Ανθρώπου – Υπολογιστή, η Φ.Ε.Γ. ασχολείται με το να επιτρέπει στους υπολογιστές να αντλούν και να ερμηνεύουν την ανθρώπινη φυσική γλώσσα. Πρόσφατη δουλειά στη Φ.Ε.Γ. είναι αλγόριθμοι βασισμένοι στη Μ.Μ. και πιο συγκεκριμένα στη στατιστική Μ.Μ. Το υψηλότερο επίπεδο γενικής ανάπτυξης εφαρμογών στη Φ.Ε.Γ. (state of the art) περιλαμβάνει κατηγοριοποίηση κειμένου, εξαγωγή πληροφοριών, ανάλυση συναισθημάτων, μηχανική μετάφραση και εφαρμόζεται σε πολλές διαφορετικές επιστημονικές περιοχές. Στη συνέχεια περιγράφονται προσεγγίσεις ανάλυσης συναισθημάτων που έχουν εφαρμοστεί σε Ε.Π.Β.

### Ανάλυση Συναισθημάτων (Α.Σ.)

Η Α.Σ. ή εξόρυξη γνώμης είναι η χρήση της Φ.Ε.Γ., η ανάλυση κειμένου και η υπολογιστική γλωσσολογία, για τον εντοπισμό και την εξαγωγή πληροφοριών ή χαρακτηριστικών από κείμενα. Στην τρέχουσα κατάσταση, η αυτοματοποιημένη Α.Σ. δεν είναι σε θέση να είναι τόσο ακριβής όσο η ανθρώπινη ανάλυση. Οι αυτοματοποιημένοι μέθοδοι Α.Σ. δεν γνωρίζουν που βρίσκονται οι λεπτές γραμμές του περιβάλλοντος, η ειρωνεία, η γλώσσα του ανθρώπινου σώματος ή ο τόνος. Στην ανθρώπινη ανάλυση, η αξιοπιστία μεταξύ βαθμολογιών παίζει μεγάλο ρόλο, και η οποία είναι ο βαθμός συμφωνίας μεταξύ των κριτών. Σύμφωνα με πρόσφατες μελέτες, το ποσοστό ανθρώπινης συμφωνίας στην ανάλυση του συναισθήματος είναι γύρω στο 79–80 %. Κανένας Ε.Π.Β. σε επιχειρηματική μορφή δεν εξετάζει οποιαδήποτε από τις προαναφερθείσες πτυχές και συνεπώς δεν θα συζητηθούν περαιτέρω.

### Εποπτευόμενη Μηχανική Μάθηση

Η Εποπτευόμενη Μηχανική Μάθηση αποσκοπεί στην πρόβλεψη συνόλων δεδομένων εξόδου ( $y_1, y_2, y_3, \dots, y_n$ ) από σύνολα δεδομένων εισόδου ( $x_1, x_2, x_3, \dots$ )

χη), για 'n' παρατηρήσεις. Μια γενική λειτουργία Μ.Μ. δημιουργείται για πρόβλεψη εξόδου από την είσοδο η οποία δεν είναι κομμάτι της εκπαίδευσης. Οι προβλέψεις προκύπτουν από ένα σύνολο εκπαιδευτικών δεδομένων, μέσω ενός γνωστού σετ εισόδου και εξόδου.

#### Μη Εποπτευόμενη Μηχανική Μάθηση

Η Μη Εποπτευόμενη Μηχανική Μάθηση είναι η διαδικασία ταξινόμησης των δεδομένων χωρίς πρόσβαση στα επισημασμένα δεδομένα εκπαίδευσης. Χρησιμοποιώντας 'n' παρατηρήσεις δεδομένων ( $x_1, x_2, x_3, \dots, x_n$ ) ο πρωταρχικός στόχος της μεθόδου αυτής είναι η συγκέντρωση δεδομένων με παρόμοιες ιδιότητες και σχέσεις, σε διαφορετικές ομάδες. Δεδομένου ότι η ετικέτα δεδομένων δεν παρέχεται, η συγκεκριμένη μέθοδος συνήθως απαιτεί μεγαλύτερο ποσό δεδομένων εκπαίδευσης για να αποδώσει τόσο καλά όσο η μέθοδος της Εποπτευόμενης Μηχανικής Μάθησης.

#### Τεχνητά Νευρωνικά Δίκτυα (Τ.Ν.Δ. – ANN's)

Στη Μ.Μ. και τη γνωστική επιστήμη, Τεχνητά Νευρωνικά Δίκτυα (ANN's – Artificial Neural Networks) είναι μια οικογένεια μοντέλων, εμπνευσμένων από βιολογικά νευρωνικά δίκτυα και συγκεκριμένα από τον ανθρώπινο εγκέφαλο. Τα τεχνητά νευρωνικά δίκτυα συνήθως είναι κατασκευασμένα σε επίπεδα, όπου κάθε ένα από αυτά παίζει ένα συγκεκριμένο ρόλο και περιέχει έναν αριθμό τεχνητών νευρώνων. Συνήθως, αυτά τα επίπεδα είναι η είσοδος, η έξοδος και τα πολλαπλά επίπεδα ανάμεσά τους. Ο πραγματικός υπολογισμός, η επεξεργασία και η στάθμιση των νευρώνων γίνεται μέσω των κρυμμένων επιπέδων και είναι καθοριστικής σημασίας για την απόδοση του δικτύου. Μπορεί να θεωρηθεί ότι οι περισσότερες οντότητες και οι Ε.Π.Β. χρησιμοποιούν διάφορα Νευρωνικά Δίκτυα ή και συνδυασμό αυτών με άλλους τύπους τεχνικών. Τα ANN's προτιμώνται λόγω της ικανότητάς τους να αντιμετωπίζουν μη γραμμικές σχέσεις, ασαφείς καθώς και μη επαρκή δεδομένα, αλλά και της ικανότητάς τους να μαθαίνουν και να προσαρμόζονται στις αλλαγές σε σύντομο χρονικό διάστημα.

### Μέθοδοι

Περιγράφονται οι χρησιμοποιούμενες ερευνητικές μέθοδοι και οι μέθοδοι συλλογής δεδομένων καθώς και οι χρησιμοποιούμενες για την ανάλυση συναισθημάτων και δεδομένων.

### Μελέτη Βιβλιογραφίας

Μέσω της έρευνας σε ακαδημαϊκά άρθρα, ψηφιακά άρθρα, έγγραφα και βιβλία στην περιοχή της Τ.Ν., της Μ.Μ. και της Επιστήμης Υπολογιστών, οι θεωρητικές αρχές του πεδίου των Ε.Π.Β. έχουν καλυφθεί.

### **Έξυπνοι Βοηθοί**

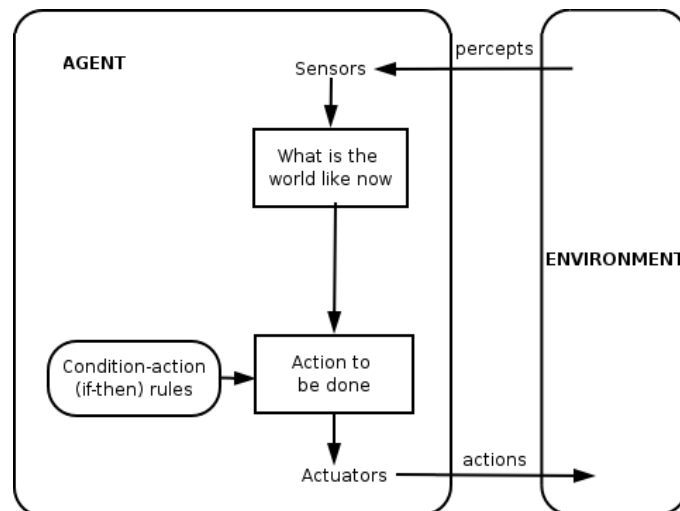
Στην τεχνητή νοημοσύνη, ένας Intelligent Agent<sup>2</sup> (Έξυπνος Βοηθός) είναι μια αυτόνομη οντότητα η οποία παρατηρεί μέσω αισθητήρων και δρώντας πάνω σε κάποιο περιβάλλον χρησιμοποιώντας ενεργοποιητές, κατευθύνει τη δραστηριότητά του στην επίτευξη στόχων. Θα μπορούσε να πει κανείς ότι κατά κάποιο τρόπο είναι ορθολογιστική η οντότητα αυτή. Οι έξυπνοι βοηθοί μπορούν επίσης να μάθουν με την χρήση της μηχανικής μάθησης<sup>3</sup> (machine learning) ή να χρησιμοποιήσουν την γνώση τους για την επίτευξη των στόχων. Μπορεί να είναι απλοί ή σύνθετοι. Ένα αντανακλαστικό μηχάνημα, όπως ένας θερμοστάτης θεωρείται ένα παράδειγμα ενός Έξυπνου Βοηθού. Οι Έξυπνοι Βοηθοί συχνά περιγράφονται σχηματικά σαν ένα αφηρημένο λειτουργικό σύστημα παρόμοιο με το πρόγραμμα ενός υπολογιστή. Για το λόγο αυτό, οι Έξυπνοι Βοηθοί μερικές φορές λέγονται **«Αφηρημένοι Έξυπνοι Βοηθοί»**, για να ξεχωρίζουν από τις πραγματικές παγκόσμιες υλοποιήσεις όπως συστήματα υπολογιστών, βιολογικά συστήματα ή οργανισμούς. Από την άλλη, κάποιοι ορισμοί των Έξυπνων Βοηθών τονίζουν την αυτονομία τους, και έτσι λοιπόν προτιμάται η έννοια **«Αυτόνομοι Έξυπνοι Βοηθοί»**.

---

<sup>2</sup> [https://en.wikipedia.org/wiki/Intelligent\\_agent](https://en.wikipedia.org/wiki/Intelligent_agent)

<sup>3</sup> [https://en.wikipedia.org/wiki/Machine\\_learning](https://en.wikipedia.org/wiki/Machine_learning)

Όμως υπάρχουν και αυτοί που θεωρούνται, όσον αφορά την ουσία της νοημοσύνης, ότι κατευθύνονται σύμφωνα με το στόχο, κι έτσι προτιμάται μια ορολογία δανεισμένη από τα οικονομικά, η οποία λέγεται «**Λογικοί Βοηθοί**». Οι Έξυπνοι Βοηθοί στην τεχνητή νοημοσύνη είναι στενά συνδεδεμένοι με τους «**Βοηθούς Λογισμικού**» (Software Agents), ένα αυτόνομο υπολογιστικό πρόγραμμα που εκτελεί εργασίες για χάρη των χρηστών. Στην επιστήμη των υπολογιστών, ο όρος «Έξυπνος Βοηθός» μπορεί να χρησιμοποιηθεί σαν αναφορά σε έναν «Βοηθό Λογισμικού» ο οποίος εμφανίζει κάποια ευφυΐα, ανεξάρτητα με το αν δεν ανήκει στην κατηγορία των «Λογικών Βοηθών». Για παράδειγμα, αυτόνομα προγράμματα που χρησιμοποιούνται στην εξόρυξη δεδομένων (κάποιες φορές λέγονται bots) ή για βοήθεια του χειριστή, επίσης λέγονται «Έξυπνοι Βοηθοί». Στο παρακάτω διάγραμμα φαίνεται ένας απλός αντανακλαστικός βοηθός.



*Διάγραμμα 1.4. Simple Reflex Agent*

Οι «έξυπνοι βοηθοί» (Intelligent Agents), είναι ένα λογισμικό το οποίο μπορεί να εκτελέσει έργα ή υπηρεσίες για λογαριασμό ενός ανθρώπου. Μπορούν να έχουν πρόσβαση σε πληροφορίες από πολλές online πηγές όπως τις καιρικές συνθήκες, την κίνηση στους δρόμους, τα Νέα και άλλα. Έχουν επίσης την ικανότητα να οργανώνουν και να διατηρούν πληροφορίες όπως τη διαχείριση των e-mails, τα

συμβάντα του ημερολογίου, τα αρχεία κλπ. Κάποιοι έξυπνοι βοηθοί μπορούν να παρέχουν πληροφορίες μέσω φωνητικών εντολών. Κάποιοι από τους δημοφιλέστερους έξυπνους Βοηθούς, είναι οι εξής:

- SIRI
- GoogleNow
- Cortana
- Jasper
- Open Jarvis
- Kalliope
- Sonus

#### **Πιο αναλυτικά:**

##### **SIRI<sup>4,5</sup>:**

Χρησιμοποιεί φωνητικά ερωτήματα και ένα είδος ανθρώπινου υπολογιστικού περιβάλλοντος (Natural Language User Interfaces) έτσι ώστε να κάνει ερωτήσεις, να προτείνει πράγματα και να εκτελεί ενέργειες<sup>6</sup>. Το λογισμικό προσαρμόζεται στη χρήση των προσωπικών γλωσσών, στις αναζητήσεις και στις προτιμήσεις των χρηστών, με συνεχόμενη χρήση. Τα αποτελέσματα που επιστρέφει είναι εξατομικευμένα. Ανήκει στην Apple Inc. Αρχικά αναπτύχθηκε από το SRI International Artificial Intelligence Center. Η μηχανή αναγνώρισης ομιλίας του SIRI<sup>7</sup> παρέχεται από την Nuance Communications.

##### **Google Now<sup>8</sup>:**

Το Google Now είναι ένας έξυπνος βοηθός που αναπτύχθηκε από την εταιρεία Google. Είναι διαθέσιμο στις εφαρμογές Google για λογισμικό Android και iOS.

---

<sup>4</sup> <https://en.wikipedia.org/wiki/Siri>

<sup>5</sup> <https://www.macrumors.com/guide/siri/>

<sup>6</sup> <https://www.imore.com/siri>

<sup>7</sup> <https://www.apple.com/siri/>

<sup>8</sup> [https://en.wikipedia.org/wiki/Google\\_Now](https://en.wikipedia.org/wiki/Google_Now)

Χρησιμοποιεί όπως και το SIRI ένα είδος ανθρώπινου υπολογιστικού περιβάλλοντος (Natural Language User Interfaces) για την απάντηση ερωτήσεων κλπ. Μαζί με το να απαντάει τις ερωτήσεις των χρηστών, το Google Now προληπτικά εκφωνεί πληροφορίες τις οποίες προβλέπει πως οι χρήστες θέλουν να δουν, με βάση το τι έχουν ψάξει. Παλαιότερα ενεργοποιόταν λέγοντας “Okay Google Now”, είτε από τις εφαρμογές Google ή την οθόνη του τηλεφώνου. Επί του παρόντος ενεργοποιείται απλώς λέγοντας “Okay Google”.

Συμπεριλήφθηκε για πρώτη φορά στην έκδοση Android 4.1. Είναι ανταγωνιστής του SIRI της Apple (που αναφέρθηκε ως άνω) και του Cortana της Microsoft (που θα ακολουθήσει στη συνέχεια).

Σε αντίθεση με το SIRI, του οποίου οι πηγές προκύπτουν από συγκεκριμένες ιστοσελίδες, οι πηγές του Google Now προέρχονται από οποιαδήποτε σχετική ιστοσελίδα. Από την άποψη της αναγνώρισης ομιλίας και από διάφορες μελέτες δεν έχει ξεκάθαρα αποδειχθεί πιο από τα δύο είναι περισσότερο ακριβές.

#### **Cortana<sup>9</sup>:**

Το Cortana είναι και αυτό ένας έξυπνος βοηθός που δημιουργήθηκε όμως από την Microsoft για τις εξής πλατφόρμες, και σύντομα για το Amazon Alexa:

- Microsoft Windows 10
- Windows 10 Mobile
- Windows Phone 8.1
- Invoke Smart Speaker
- Microsoft Band
- Xbox One
- iOS
- Android
- Windows Mixed Reality

---

<sup>9</sup> <https://en.wikipedia.org/wiki/Cortana>

Το Cortana<sup>10</sup> μπορεί να θέτει υπενθυμίσεις, να αναγνωρίζει φυσική φωνή χωρίς να χρειάζεται να πληκτρολογηθεί κάτι και να απαντάει ερωτήματα αντλώντας πληροφορίες από την μηχανή αναζήτησης Bing.

Είναι διαθέσιμο στα Αγγλικά, Ιαπωνικά, Κινέζικα και πολλές άλλες γλώσσες, ανάλογα με την πλατφόρμα λογισμικού και την περιοχή που βρίσκεται. Κυρίως ανταγωνίζεται το SIRI, όλο το Google Assistant συμπεριλαμβανομένου και του Google Now, και το Amazon Alexa.

Πρώτη φορά έγινε επίδειξη του Cortana στη διάσκεψη “Microsoft BUILD Developer”. Λανσαρίστηκε, σαν βασικό συστατικό του σχεδίου «αλλαγής» της Microsoft, των μελλοντικών λειτουργικών συστημάτων για τα Windows και τα Windows Phone.

Η εξέλιξη του Cortana ξεκίνησε το 2009. Οι ικανότητες φυσικής επεξεργασίας γλώσσας συμπληρώνονται από το “Tell me Net works” και είναι συζευγμένες με μια βάση δεδομένων που ονομάζεται Satori.

### **Open Jarvis<sup>11</sup>:**

Το Open Jarvis είναι μια ελαφριά, πολυγλωσσική εφαρμογή που μπορεί να ρυθμιστεί (είναι δηλαδή configurable). Σκοπεύει στον αυτοματισμό σπιτιού και τρέχει σε μικροϋπολογιστή, όπως το Raspberry Pi 2 και 3. Εγκαθιστά αυτόματα μηχανές μετατροπής ομιλίας σε κείμενο (όπως τα Pocket Sphinx, Snow Boy, Wit) και μηχανές μετατροπής κειμένου σε ομιλία (όπως το eSpeak, SVOX Pico). Υπάρχουν plug-ins για να προσθέσει κανείς ωραία χαρακτηριστικά. Υποστηρίζει επίσης, λειτουργίες που έχουν υλοποιηθεί από τρίτους (3<sup>rd</sup> Party Modules), κάποιες από τις οποίες είναι το NoteRed, το GUI, το Cron κ.α.

---

<sup>10</sup> <https://www.microsoft.com/en-us/cortana>

<sup>11</sup> <https://github.com/sukeesh/Jarvis>

### **Kalliope<sup>12</sup>**

Το Kalliope είναι ένας έξυπνος βοηθός που λειτουργεί μέσω φωνητικών εντολών, σχεδιασμένος για αυτοματισμό σπιτιού και για έξυπνα περιβάλλοντα. Μπορεί να αναπτύξει κανείς δικές του μεθόδους επικοινωνίας με άλλες εφαρμογές ή να χρησιμοποιήσει έτοιμες από τις ήδη υπάρχουσες. Βασική παρατήρηση είναι ότι ο συγκεκριμένος έξυπνος βοηθός είναι ανεπτυγμένος στα Γαλλικά.

### **Sonus<sup>13</sup>:**

Το Sonus είναι μια βιβλιοθήκη ομιλίας σε κείμενο με προσαρμόσιμη ανίχνευση δομοστοιχείων ακουστικής ακρόασης. Μέσω της βιβλιοθήκης αυτής, μπορεί κανείς γρήγορα και εύκολα να προσθέσει ένα φωνητικό περιβάλλον χρήστη (Voice User Interface) σε οποιοδήποτε hardware ή software project. Όπως το Google Now, το Siri και το Alexa, το Sonus λειτουργεί και off-line. Όταν ακούσει την λέξη – κλειδί, η ομιλία μεταδίδεται στην υπηρεσία αναγνώρισης σύννεφων που έχουν επιλεγεί και τότε δίνει τα αποτελέσματα.

---

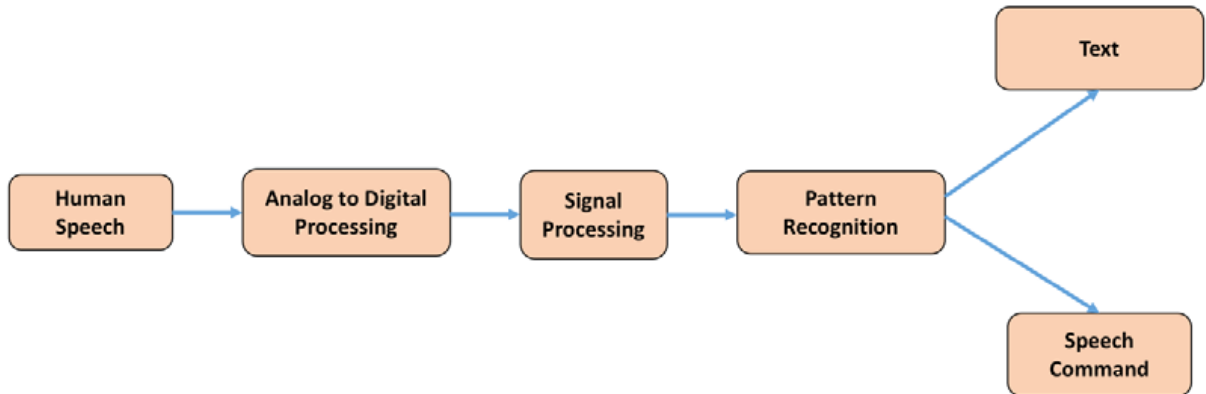
<sup>12</sup> <https://kalliope-project.github.io/>

<sup>13</sup> <https://github.com/evancohen/sonus>



### 1.3. Αναγνώριση προτύπων για την τεχνολογία ομιλίας

Η αναγνώριση προτύπων αποτελεί επιστημονικό πεδίο στη μηχανική μάθηση και είναι η βασική γραμμή για την αναγνώριση ομιλίας. Γενικά, είναι δυνατόν να κατασκευαστεί ένα σύστημα αναγνώρισης ομιλίας βάσει του παρακάτω σχεδίου:



*Διάγραμμα 1.3. Σύστημα Αναγνώρισης Ομιλίας*

Από τη μορφή ανθρώπινης ομιλίας, πρέπει να μετατραπεί σε ψηφιακή μορφή, δηλαδή σε διακριτά δεδομένα, με την εφαρμογή κάποιων μεθόδων επεξεργασίας σήματος, όπως η “αφαίρεση” του θορύβου (από τα διακριτά δεδομένα). Στην αναγνώριση προτύπων, εκτελείται μέθοδος αναγνώρισης ομιλίας. Ερευνητές έχουν πετύχει κάποιες προσεγγίσεις με χρήση υπολογιστών, χρησιμοποιώντας το Hidden Markov Model (HMM) ούτως ώστε να αναγνωρίσουν ήχους που σχετίζονται με λέξεις. Το να εκτελείται εξαγωγή χαρακτηριστικών σε ψηφιακά δεδομένα ομιλίας, είναι ένα μέρος των δραστηριοτήτων αναγνώρισης προτύπων.

## ΚΕΦΑΛΑΙΟ ΔΕΥΤΕΡΟ

### 2. ΜΙΚΡΟΪΠΟΛΟΓΙΣΤΕΣ ΚΑΙ EMBEDDED DEVICES

#### 2.1. Raspberry Pi και άλλες επιλογές Μικροϋπολογιστών

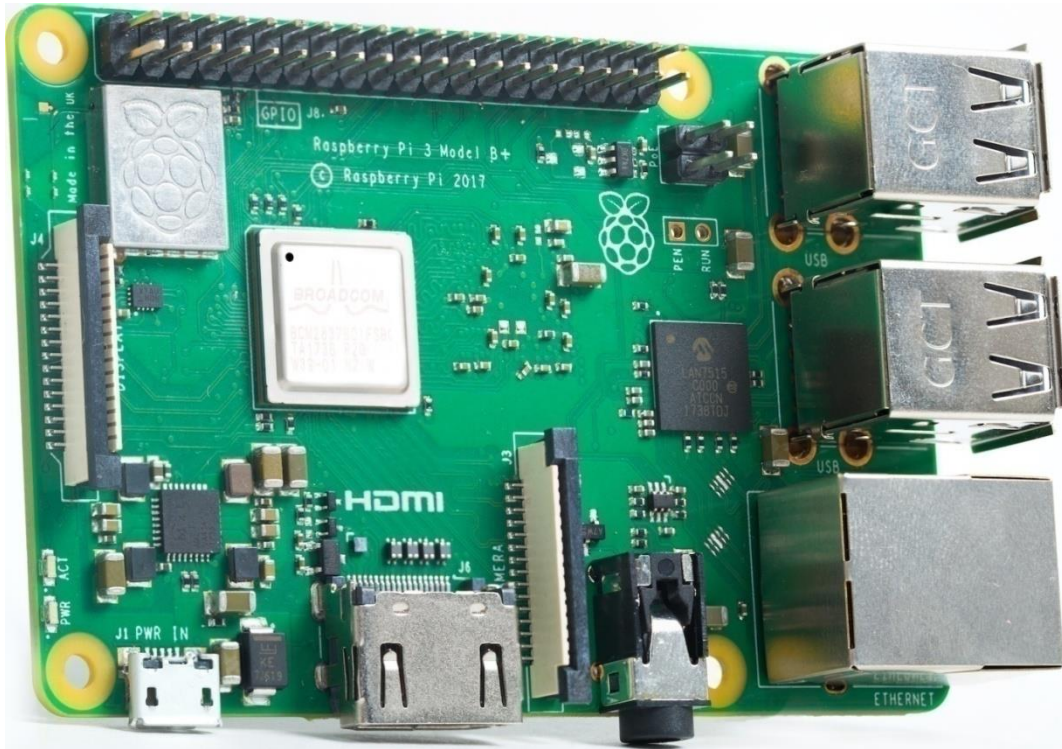
Το Raspberry Pi 3 Model B<sup>14</sup> είναι ένας μικροϋπολογιστής, δηλαδή ένας κανονικός υπολογιστής, σε μέγεθος πιστωτικής κάρτας. Είναι από τους πλέον διαδεδομένους στην Παγκόσμια Αγορά. Κοστίζει περίπου 40 ευρώ. Σχεδιάστηκε για να βοηθήσει τους νέους ανθρώπους να μάθουν να προγραμματίζουν, ενώ σαν συσκευή είναι ικανή να επιτελέσει πολλές εντυπωσιακές εργασίες. Τη δεκαετία του '80, τα παιδιά έπρεπε να γνωρίζουν να προγραμματίζουν έναν Η/Υ για να μπορέσουν να τον χρησιμοποιήσουν και αυτό είχε σαν αποτέλεσμα τα παιδιά αυτά να μεγαλώσουν αντιλαμβανόμενα το πως λειτουργούν οι υπολογιστές. Στη σημερινή εποχή, η ανάγκη για προγραμματιστές είναι ακόμη μεγαλύτερη. Έτσι, για να αντιμετωπιστεί αυτό το πρόβλημα, κάποιοι επινόησαν το Raspberry Pi. Χρησιμοποιεί μια Linux έκδοση, το Raspbian<sup>15</sup>, ένα ελεύθερο λειτουργικό σύστημα από μια κάρτα SD, όπως εκείνες που χρησιμοποιούνται στις φωτογραφικές μηχανές και τροφοδοτείται με ρεύμα. Μπορεί να τροφοδοτηθεί ακόμη και από έναν απλό φορτιστή USB κινητού τηλεφώνου. Το μόνο που έχει να κάνει κανείς, είναι να συνδέσει ένα ποντίκι, ένα πληκτρολόγιο και μια οθόνη. Κάθε μαθητής στο σχολείο, με τη χρήση του Raspberry Pi μπορεί όχι μόνο να μάθει προγραμματισμό, αλλά να το αξιοποιήσει με διαθεματικό τρόπο σε πολλά αντικείμενα και εφαρμογές. Από την πρώτη στιγμή που κυκλοφόρησε το Raspberry Pi, έχουν γίνει ευρέως γνωστά πολλά παραδείγματα χρήσης, του με ποικίλους και εντυπωσιακούς τρόπους, εκμεταλλευόμενοι οι χρήστες κάθε φορά, το μέγεθος, τη φορητότητα, το κόστος και τις πλούσιες

---

<sup>14</sup> <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>

<sup>15</sup> <https://www.raspberrypi.org/downloads/raspbian/>

δυνατότητες προγραμματισμού και συνδεσιμότητάς του. Μια παρουσίαση του μοντέλου Raspberry Pi 3 B+ εμφανίζεται παρακάτω:



Εικόνα 1: Raspberry Pi 3 Model B+

## 2.2. Ανασκόπηση Modules ήχου και ομιλίας για IoT συσκευές

Υπάρχουν διάφορα modules (δομοστοιχεία) τα οποία μπορούν να ενσωματωθούν σε πινακίδες μικροελεγκτών (MCU Boards)<sup>16</sup>. Υπάρχουν πολλά δομοστοιχεία που σχετίζονται με την επεξεργασία ομιλίας και ήχου. Κάθε ένα από αυτά, έχει τα δικά του χαρακτηριστικά και εξυπηρετεί αναλόγως με το που σκοπεύει κάποιος. Ένα παράδειγμα δομοστοιχείων ήχου και ομιλίας είναι το **Easy VR 3**<sup>17</sup> και το Easy VR Shield 3, της Veear.

Υποστηρίζονται αρκετές γλώσσες για αυτό, όπως Αγγλικά, Ιταλικά, Ισπανικά, Ιαπωνικά και Γερμανικά. Παρακάτω εικονίζεται η πολυλειτουργική μονάδα αναγνώρισης ομιλίας Easy VR 3:

<sup>16</sup> <https://en.wikipedia.org/wiki/Microcontroller>

<sup>17</sup> <https://www.sparkfun.com/products/15453>



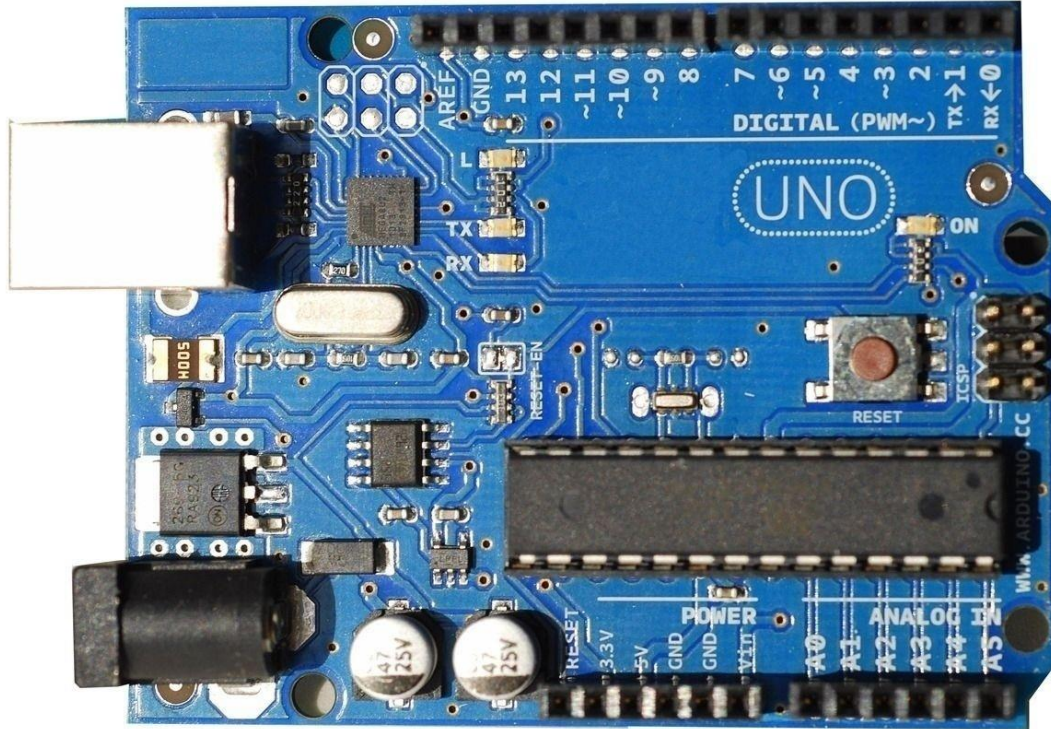
Εικόνα 2: Μονάδα Easy VR 3

Η μονάδα αυτή, έχει σχεδιαστεί για ευέλικτες, ανθεκτικές και οικονομικά αποδοτικές δυνατότητες αναγνώρισης ομιλίας και φωνής για κάθε εφαρμογή. Υποστηρίζει 32 εντολές ομιλίας, οι οποίες εξαρτώνται από το χρήστη και 26 εντολές που είναι ανεξάρτητες από αυτόν.

Ένα άλλο παράδειγμα, είναι το **Arduino**<sup>18</sup>, μια ευρέως διαδεδομένη αναπτυξιακή πλατφόρμα. Το μεγαλύτερο μέρος των Arduino χρησιμοποιεί τον μικροελεγκτή Atmel AVR. Μπορεί να γίνει χρήση πλακετών που λέγονται Arduino Shields για επέκταση δυνατοτήτων, όπως Bluetooth, Wi-Fi, κ.α. Το πλεονέκτημα μιας τέτοιας χρήσης είναι η δυνατότητα εστίασης στον προγραμματισμό της πλατφόρμας και η παράκαμψη θεμάτων σχετικών με το hardware. Το μόνο που

<sup>18</sup> <https://www.arduino.cc/>

πρέπει να κάνει κανείς, είναι η σύνδεση της Arduino Shield με το Arduino. Ακολουθως εικονίζεται μια τέτοια πλακέτα :



Εικόνα 3: Arduino UNO R3 Compatible Board

Ένα ακόμη παράδειγμα, είναι η μονάδα **Emic-2**<sup>19</sup>. Επιτρέπει τη δυνατότητα μετατροπής κειμένου σε ομιλία σε διάφορες ενσωματωμένες υπολογιστικές εφαρμογές. Για τη λειτουργία απαιτείται σύνδεση σε τροφοδοσία 5VDC, ένα ηχείο στην έξοδο ηχείου (audio output) ή υποδοχή ακουστικών και αποστολή σειριακής ροής κειμένου στα 9600 bps. Η μονάδα περιέχει όλα τα στοιχεία που είναι απαραίτητα για την ανάλυση του κειμένου σε ομιλία ώστε στη συνέχεια να δημιουργήσει φυσική ηχητική ομιλία. Αυτό που πρέπει να κάνει ο ελεγκτής είναι να στείλει σειριακές συμβολοσειρές.

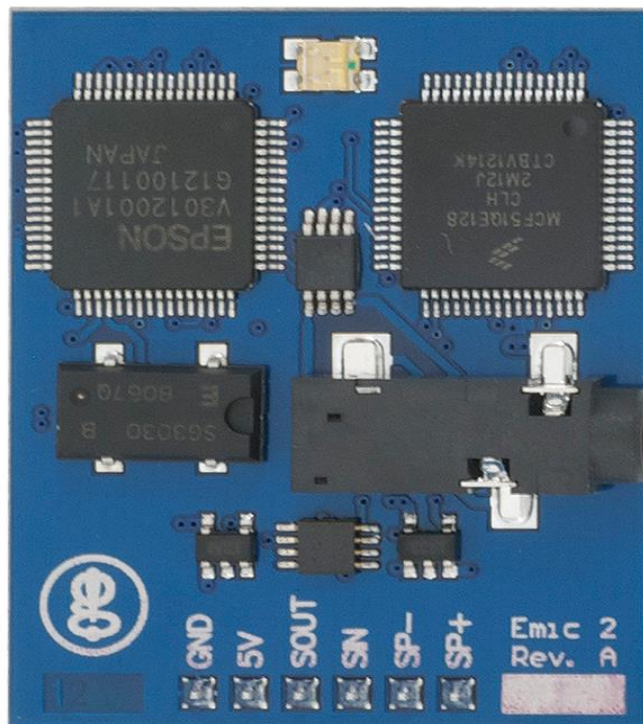
Υποστηρίζει 2 γλώσσες (Αγγλικά και Ισπανικά), και μπορεί κανείς να επιλέξει μεταξύ 9 διαφορετικών φωνών τροποποιώντας τις παραμέτρους ομιλίας κατά την εκτέλεση. Το Emic-2 βασίζει την λειτουργία του στη μηχανή μετατροπής κειμένου σε ομιλία DECTalk.

<sup>19</sup> <http://www.grandideastudio.com/emic-2-text-to-speech-module/>

Τα χαρακτηριστικά της πλακέτας αυτής είναι τα εξής:

- υψηλής ποιότητας σύνθεση ομιλίας για αγγλική και ισπανική γλώσσα,
- εννέα προκαθορισμένες φωνές περιλαμβανομένων της ανδρικής, γυναικείας και παιδικής,
- δυναμικός έλεγχος των χαρακτηριστικών ομιλίας και φωνής, συμπεριλαμβανομένης της τονικότητας, του ρυθμού ομιλίας και της έμφασης λέξης,
- βιομηχανικός πρότυπος μηχανισμός συνθέσεως λέξεων-ομιλίας DECtalk (5.0.E1),
- ενσωματωμένος ενισχυτής ισχύος ήχου και υποδοχή ήχου 1/8" (3,5 mm),
- μονόγραμμη, 6 ακίδων, κεφαλίδα 0,1" για εύκολη σύνδεση με ένα σύστημα κεντρικού υπολογιστή.

Μια τέτοια πλακέτα εμφανίζεται στην εικόνα 4:



Εικόνα 4: Emic 2 Sound-To-Text Module

### 2.3. State of the art: Speech Recognition

Η **αναγνώριση ομιλίας** είναι ένα διεπιστημονικό υποπεδίο υπολογιστικής γλωσσολογίας που αναπτύσσει μεθοδολογίες και τεχνολογίες που επιτρέπουν την αναγνώριση και τη μετάφραση της ομιλούμενης γλώσσας σε κείμενο από τους υπολογιστές. Είναι επίσης γνωστή, ως **αυτόματη αναγνώριση ομιλίας (ASR- Automatic Speech Recognition)**, **αναγνώριση ομιλίας υπολογιστή ή ομιλία σε κείμενο (STT- Speech-to-Text)**. Ενσωματώνει γνώσεις και έρευνες στους τομείς της γλωσσολογίας, της πληροφορικής και της ηλεκτρολογίας.

Ορισμένα συστήματα αναγνώρισης ομιλίας απαιτούν «εκπαίδευση», κατά την οποία ένας μεμονωμένος ομιλητής διαβάζει κείμενο ή απομονωμένο λεξιλόγιο στο σύστημα. Το σύστημα αναλύει τη συγκεκριμένη φωνή του ατόμου και το χρησιμοποιεί για να τελειοποιήσει την αναγνώριση της ομιλίας αυτού του ατόμου, με αποτέλεσμα να υπάρχει μεγαλύτερη ακρίβεια. Τα συστήματα που δεν χρησιμοποιούν εκπαίδευση ονομάζονται συστήματα "ανεξάρτητα από ομιλητή". Τα συστήματα που απαιτούν τη λειτουργία της «εκπαίδευσης» ονομάζονται "εξαρτώμενα από ομιλητή".

Από την άποψη της τεχνολογίας, η αναγνώριση ομιλίας έχει μακρά ιστορία με πολλά κύματα σημαντικών καινοτομιών. Πιο πρόσφατα, ο τομέας έχει επωφεληθεί από τις προόδους της μηχανικής/βαθιάς μάθησης. Η πρόοδος αποδεικνύεται όχι μόνο από την αύξηση των ακαδημαϊκών εφημερίδων που δημοσιεύθηκαν και δημοσιεύονται στον τομέα, αλλά κυρίως από την υιοθέτηση εκ μέρους της παγκόσμιας βιομηχανίας μιας ποικιλίας μεθόδων βαθιάς μάθησης κατά το σχεδιασμό και την ανάπτυξη συστημάτων αναγνώρισης ομιλίας.

#### 2.4. State of the art: Speech Synthesis

Η **σύνθεση του λόγου**<sup>20</sup> είναι η τεχνητή παραγωγή της ανθρώπινης ομιλίας. Ένα σύστημα υπολογιστή που χρησιμοποιείται για το σκοπό αυτό ονομάζεται «υπολογιστής ομιλίας» ή «συνθέτης ομιλίας» και μπορεί να εφαρμοστεί σε προϊόντα λογισμικού ή υλικού. Ένα **σύστημα TTS (Text-to-speech)** μετατρέπει το κανονικό κείμενο σε ομιλία.

Άλλα συστήματα καθιστούν συμβολικές γλωσσικές αναπαραστάσεις όπως οι φωνητικές μεταγραφές στην ομιλία. Η σύνθετη ομιλία μπορεί να δημιουργηθεί συνδυάζοντας κομμάτια εγγραφής που είναι αποθηκευμένα σε μια βάση δεδομένων.

Τα συστήματα διαφέρουν στο μέγεθος των αποθηκευμένων μονάδων ομιλίας. Για συγκεκριμένους τομείς χρήσης, η αποθήκευση ολόκληρων λέξεων ή προτάσεων επιτρέπει την παραγωγή υψηλής ποιότητας. Εναλλακτικά, ένας συνθέτης μπορεί να ενσωματώσει ένα μοντέλο της φωνητικής οδού και άλλα χαρακτηριστικά ανθρώπινης φωνής για να δημιουργήσει μια απολύτως «συνθετική» έξοδο φωνής.

Η ποιότητα ενός συνθέτη ομιλίας κρίνεται από την ομοιότητά του με την ανθρώπινη φωνή και από την ικανότητά του να κατανοείται απολύτως ξεκάθαρα. Ένα κατανοητό πρόγραμμα ανάγνωσης κειμένου σε ομιλία επιτρέπει σε άτομα με προβλήματα όρασης ή αναπηρίες ανάγνωσης, να ακούν γραπτές λέξεις σε έναν οικιακό υπολογιστή. Πολλά λειτουργικά συστήματα υπολογιστών έχουν συμπεριλάβει συνθέτες ομιλίας από τις αρχές της δεκαετίας του 1990.

Ένα σύστημα κειμένου σε ομιλία, ή «μηχανή» όπως θα φανεί παρακάτω, αποτελείται από δύο μέρη: ένα εμπρός και ένα πίσω μέρος:

Το **front-end** έχει δύο σημαντικά καθήκοντα: πρώτον, μετατρέπει το ακατέργαστο κείμενο που περιέχει σύμβολα όπως αριθμούς και συντμήσεις στο

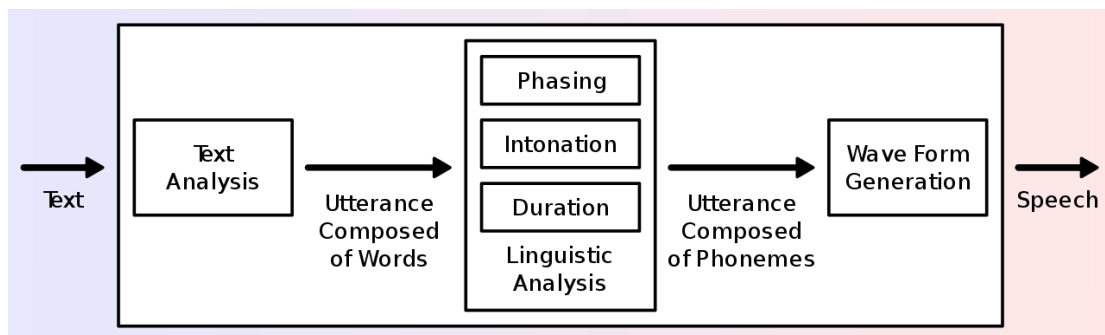
---

<sup>20</sup> [https://en.wikipedia.org/wiki/Speech\\_synthesis](https://en.wikipedia.org/wiki/Speech_synthesis)



ισοδύναμο των γραμμένων λέξεων. Αυτή η διαδικασία ονομάζεται συχνά εξομάλυνση κειμένου, προεπεξεργασία ή tokenization. Το μπροστινό μέρος εκχωρεί τότε φωνητικές μεταγραφές σε κάθε λέξη, διαιρεί και επισημαίνει το κείμενο σε φράσεις, ρήτρες και προτάσεις. Η διαδικασία εκχώρησης φωνητικών μεταγραφών σε λέξεις ονομάζεται μετατροπή κειμένου σε φωνή ή γραφή-προς-φωνή. Οι φωνητικές μεταγραφές και οι πληροφορίες μαζί, αποτελούν τη συμβολική γλωσσική αναπαράσταση που εξάγεται από το front-end.

Το **back-end** -που συχνά αναφέρεται ως συνθεσάιζερ- μετατρέπει στη συνέχεια τη συμβολική γλωσσική αναπαράσταση στον ήχο. Σε ορισμένα συστήματα, αυτό το μέρος περιλαμβάνει τον υπολογισμό του contour pitch ή phoneme durations, που στη συνέχεια επιβάλλεται στην ομιλία εξόδου. Παρακάτω μπορεί να δει κάποιος ένα τυπικό διάγραμμα ενός συστήματος μετατροπής κειμένου-σε-ομιλία.



Διάγραμμα 2.1. Λογική Text-To-Speech Μετατροπής Συστήματος

## 2.5. State of the art: Chatbots

Ένα chatbot<sup>21</sup> είναι ένα κομμάτι λογισμικού που διεξάγει μια συζήτηση με τη μέθοδο ακουστικών ή κειμένων. Τέτοια προγράμματα σχεδιάζονται συχνά για να προσομοιάσουν πειστικά πώς ένας άνθρωπος θα συμπεριφερόταν ως συνομιλητικός εταίρος, παρόλο που από το 2019 δεν έχουν περάσει πολλοί από τη “δοκιμασία του Turing” (που αναπτύχθηκε για να τεστάρει την ισοδυναμία ή μη της ευφυούς συμπεριφοράς ενός μηχανήματος, με αυτή του ανθρώπου).

Το 1950 δημοσιεύθηκε το διάσημο άρθρο του Alan Turing "Computing Machinery and Intelligence"<sup>22</sup>, το οποίο πρότεινε αυτό που τώρα ονομάζεται «δοκιμασία Turing» ως κριτήριο της νοημοσύνης. Αυτό το κριτήριο εξαρτάται από την ικανότητα ενός προγράμματος υπολογιστή να μιμείται έναν άνθρωπο σε γραπτή συνομιλία σε πραγματικό χρόνο, με έναν άνθρωπο κριτή στο βαθμό που ο κριτής δεν είναι σε θέση να διακρίνει αξιόπιστα -με βάση μόνο το περιεχόμενο της συζήτησης- μεταξύ του προγράμματος και ενός πραγματικού ανθρώπου.

Τα chatbots χρησιμοποιούνται συνήθως σε συστήματα διαλόγου για διάφορους πρακτικούς σκοπούς, συμπεριλαμβανομένης της εξυπηρέτησης πελατών ή της απόκτησης πληροφοριών. Ορισμένα από αυτά τα λογισμικά χρησιμοποιούν εξελιγμένα συστήματα επεξεργασίας φυσικής γλώσσας ενώ πολλά απλούστερα, σαρώνουν λέξεις-κλειδιά μέσα στην είσοδο και στη συνέχεια τραβούν μια απάντηση με τις πιο κατάλληλες λέξεις-κλειδιά ή το πλησιέστερο πρότυπο διατύπωσης, από μια βάση δεδομένων.

Ο όρος "ChatterBot" σχεδιάστηκε αρχικά από τον Michael Mauldin (δημιουργός του πρώτου Verbot -Verbal-Robot- Julia) το 1994 για να περιγράψει αυτά τα προγράμματα συνομιλίας. Σήμερα, τα περισσότερα chatbots προσπελάζονται μέσω εικονικών βοηθών όπως το Google Assistant και το Amazon Alexa, μέσω εφαρμογών ανταλλαγής μηνυμάτων όπως το Facebook, Messenger ή το WeChat ή μέσω εφαρμογών και ιστότοπων μεμονωμένων οργανισμών. Τα

---

<sup>21</sup> <https://en.wikipedia.org/wiki/Chatbot>

<sup>22</sup> <https://academic.oup.com/mind/article/LIX/236/433/986238>

chatbots μπορούν να ταξινομηθούν σε κατηγορίες χρήσης που περιλαμβάνουν το συνομιλητικό εμπόριο (ηλεκτρονικό εμπόριο μέσω chat), την εκπαίδευση, την ψυχαγωγία, τη χρηματοδότηση, την υγεία, τις ειδήσεις και την παραγωγικότητα.

Παρά το γεγονός ότι τα chatbots έχουν κερδίσει φήμη σε μεγάλη κλίμακα (και ίσως ένα μικρό μερίδιο επικριτών τους), εκτεταμένη χρήση μοιράζεται και το Conversational AI<sup>23</sup>, προκειμένου στην επικοινωνία υπολογιστή και ανθρώπου, μέσω κειμένου και μέσω φωνής. Αναφέρεται στη χρήση εφαρμογών ανταλλαγής μηνυμάτων, βοηθών ομιλίας και chatbots για αυτοματοποίηση της επικοινωνίας και δημιουργία εξατομικευμένων εμπειριών πελατών σε κλίμακα.

Οι σχεδιαστές διασυνδέσεων έχουν κατανοήσει ότι η ετοιμότητα των ανθρώπων να ερμηνεύσουν την παραγωγή του υπολογιστή ως πραγματικά συνομιλητική -ακόμη και όταν στην πραγματικότητα βασίζεται σε μάλλον απλή αντιστοίχιση προτύπων- μπορεί να αξιοποιηθεί για χρήσιμους σκοπούς.

Οι περισσότεροι άνθρωποι προτιμούν να ασχολούνται με προγράμματα που είναι περισσότερο 'ανθρώπινα'. Αυτό δίνει σε τεχνικές τύπου chatbot, έναν πιθανώς χρήσιμο ρόλο σε διαδραστικά συστήματα που πρέπει να αποσπάσουν πληροφορίες από τους χρήστες, εφόσον οι πληροφορίες αυτές είναι σχετικά απλές και εμπίπτουν σε προβλέψιμες κατηγορίες. Έτσι, για παράδειγμα, τα συστήματα ηλεκτρονικής βοήθειας μπορούν να χρησιμοποιήσουν χρήσιμες τεχνικές chatbot για να προσδιορίσουν τον τομέα βοήθειας που χρειάζονται οι χρήστες, προσφέροντας ενδεχομένως μια πιο φιλική διεπαφή από ένα πιο επίσημο σύστημα αναζήτησης ή μενού.

---

<sup>23</sup> <https://www.interactions.com/conversational-ai/>

## ΚΕΦΑΛΑΙΟ ΤΡΙΤΟ

### 3. Ο ΕΞΥΠΝΟΣ ΒΟΗΘΟΣ JASPER

Το Jasper<sup>24</sup> είναι μια πλατφόρμα ανοικτού κώδικα για ανάπτυξη εφαρμογών ελεγχόμενων μέσω της ανθρώπινης φωνής, εξαιρετικά ελαφριά και βρίσκεται σε κατάσταση λειτουργίας “always-on”, το οποίο σημαίνει ότι πάντοτε ακούει σε εντολές, με τη δυνατότητα λήψης αυτών από αρκετά μέτρα μακριά. Μπορεί κανείς χρησιμοποιώντας την φωνή του να ρωτήσει για πληροφορίες, να ενημερώσει μια κατάσταση σε μια σελίδα κοινωνικής δικτύωσης, να «ελέγξει» το σπίτι του κλπ.

Έχει σχεδιαστεί για εφαρμογές οικιακού αυτοματισμού, αν και δεν περιορίζεται σε αυτές. Μπορεί να λειτουργήσει σε πλατφόρμες ηλεκτρονικού υπολογιστή με εξαιρετικά μικρές διαστάσεις, όπως για παράδειγμα το Raspberry Pi. Είναι 100% ανοικτού κώδικα, εξαιρετικά προσαρμόσιμο, μεταξύ άλλων μέσω της προσθήκης plug-in και μπορεί να αγοράσει κανείς τα εξαρτήματα που θα χρησιμοποιήσει για το εκάστοτε project, να χρησιμοποιήσει τα έγγραφα που υπάρχουν στο διαδίκτυο ή ακόμη καλύτερα στην επίσημη ιστοσελίδα του Jasper Project και να δημιουργήσει με τον τρόπο αυτό τα δικά του δομοστοιχεία.

Ακόμη, δίνει την δυνατότητα επιλογής διάφορων μηχανών μετατροπής ομιλίας σε κείμενο: STT (Sound-to-Text) και μετατροπής κειμένου σε ομιλία: TTS (Text-to-Sound).

---

<sup>24</sup> <http://jasperproject.github.io/>

### 3.1. Διαδικασία Εγκατάστασης και επιλογών βιβλιοθηκών

Υπάρχουν 2 τρόποι εγκατάστασης του Jasper σε πλατφόρμα Raspberry Pi.

Ο ένας είναι εγκατάσταση μέσω απευθείας αντιγραφής της εικόνας Raspbian με το Jasper ήδη εγκατεστημένο. Είναι ο πιο γρήγορος και προτεινόμενος τρόπος.

Ο άλλος, είναι με χειρωνακτική εγκατάσταση η οποία προαπαιτεί αρκετό ελεύθερο χρόνο και όρεξη.

Σχετικά με τις υποστηριζόμενες επιλογές μηχανής STT, μπορεί να χρησιμοποιηθεί το Rocketsphinx<sup>25</sup>, η οποία είναι και η γνωστότερη όλων των μηχανών μετατροπής ομιλίας σε κείμενο, ή για παράδειγμα το Google STT<sup>26</sup>. Ωστόσο, υπάρχουν και άλλες επιλογές, όπως η μηχανή Julius.

Σχετικά με την επιλογή μηχανής TTS, δηλαδή τη μηχανή μετατροπής κειμένου σε ομιλία, δίνονται προς χρήση οι eSpeak<sup>27</sup>, Festival<sup>28</sup>, Flite<sup>29</sup> ή Google TTS, αναφερόμενες ως οι πλέον δημοφιλείς μηχανές.

Επίσης, υπάρχει υποστήριξη λειτουργιών προσφερόμενες από τρίτους (3<sup>rd</sup> Party Modules). Κάποια από αυτά που μπορεί να χρησιμοποιήσει κανείς, είναι η υπολογιστική μηχανή ανάκτησης γνώσης Wolfram Alpha<sup>30</sup>, το Google Calendar, το οποίο επικοινωνεί με την υπηρεσία Google Calendar και επεξεργάζεται την σχετική πληροφορία, το Twitter το οποίο επικοινωνεί με το Twitter και επεξεργάζεται την πληροφορία που έχει δοθεί από το χρήστη και επίσης το Status το οποίο επιστρέφει πληροφορίες σχετικά με την κατάσταση του συστήματος, όπως η θερμοκρασία. Αυτά είναι κάποια από τα παραδείγματα που μπορεί να χρησιμοποιήσει κανείς ενώ υπάρχουν και άλλα όπως Selfie, Domotics, MQTT κλπ.

Ακολουθεί κατά στάδιο η απαραίτητη διαδικασία που πρέπει να ακολουθηθεί, για να μπορέσει κανείς να εγκαταστήσει τον έξυπνο βοηθό «Jasper» και να τον χρησιμοποιήσει όπως επιθυμεί. Πρώτα απ' όλα θα πρέπει να υπάρχει

---

<sup>25</sup> <https://pypi.org/project/pocketsphinx/>

<sup>26</sup> <https://cloud.google.com/speech-to-text>

<sup>27</sup> <http://espeak.sourceforge.net/>

<sup>28</sup> <http://www.cstr.ed.ac.uk/projects/festival/>

<sup>29</sup> <http://www.festvox.org/flite/>

<sup>30</sup> <https://www.wolframalpha.com/>

στη διάθεση του χρήστη ένα Raspberry Pi Model B, ή νεότερο (κατά προτίμηση, για να ελαχιστοποιηθούν περιπτώσεις ασυμβατότητας). Το «Jasper» είναι ειδικά σχεδιασμένο για το Raspberry Pi Model B, απαιτείτε όμως και μια περαιτέρω κατοχή περιφερειακών αντικειμένων. Τα απολύτως απαραίτητα είναι τα εξής (δεδομένου ότι υπάρχει ήδη ένα Raspberry Pi Model B ή νεότερο):

- I. οποιοδήποτε μικρόφωνο με USB κατάληξη,*
- II. τουλάχιστον μια κάρτα μνήμης τύπου SD, με ελάχιστη χωρητικότητα 16 GB,*
- III. καλώδιο Ethernet,*
- IV. ηχεία ή ακουστικά,*
- V. πληκτρολόγιο και προαιρετικά ποντίκι και*
- VI. τροφοδοτικό τάσης για το Raspberry Pi.*

Η συνδεσμολογία είναι αρκετά απλή. Τοποθετούνται στην πλακέτα το μικρόφωνο, η SD κάρτα, το καλώδιο του Ίντερνετ, το πληκτρολόγιο και το ποντίκι, εάν υπάρχει, τα ηχεία ή τα ακουστικά και τέλος το τροφοδοτικό. Η αρχιτεκτονική του συστήματος είναι τέτοια, ούτως ώστε όταν το τροφοδοτικό συνδεθεί, το Raspberry Pi θα πραγματοποιήσει λειτουργία εκκίνησης, οπότε καλό είναι η τροφοδοσία να συνδέεται στο τέλος.

Όπως αναφέρθηκε και νωρίτερα, υπάρχουν δύο μέθοδοι για να εγκαταστήσει κανείς το «Jasper». Σαφώς, ο πιο γρήγορος και θεωρητικά ασφαλέστερος τρόπος είναι να κατεβάσει κανείς το προεγκατεστημένο disk image το οποίο υπάρχει διαθέσιμο<sup>31</sup>. Υπάρχει επίσης ένα ανεπίσημο image για το μοντέλο B+. Αφού τερματιστεί το imaging στην SD κάρτα, δηλαδή αφού πραγματοποιηθεί η διαδικασία εγκατάστασης, θα πρέπει να κλωνοποιηθεί το repository και να εγκατασταθούν τα dependencies, τα λογισμικά εξαρτήματα δηλαδή, της Python. Στη συνέχεια περιγράφεται αναλυτικά πως γίνεται αυτό.

Είναι σχεδόν βέβαιο ότι, εάν προτιμήσει κανείς τον τρόπο που περιγράφηκε παραπάνω, θα αντιμετωπίσει προβλήματα συμβατότητας, θα περιγραφεί αναλυτικά ποιός είναι ίσως ο καταλληλότερος τρόπος και ο οποίος μας προετοιμάζει ούτως ώστε να μπορούμε να εντοπίσουμε, αλλά και να διαχειριστούμε καταστάσεις ασυμβατότητας αλλά και σφαλμάτων.

---

<sup>31</sup> <http://jasperproject.github.io/documentation/>

### **ΕΝΑΡΞΗ ΔΙΑΔΙΚΑΣΙΑΣ ΕΓΚΑΤΑΣΤΑΣΗΣ**

Πρώτα απ' όλα, πρέπει να σιγουρέψουμε πως ο διαθέσιμος χώρος στην κάρτα μνήμης μας, είναι εκμεταλλεύσιμος στο 100%. Αυτό μπορούμε να το κάνουμε με διάφορα βοηθητικά προγράμματα, αλλά ο προτεινόμενος τρόπος είναι το λεγόμενο "GPARTED", το οποίο είναι διαθέσιμο σε λογισμικό Linux. Αφού ολοκληρωθεί η διαδικασία αυτή, η οποία είναι εξαιρετικά απλή, θα χρειαστεί να εγκαταστήσουμε το λειτουργικό σύστημα του Raspberry Pi. Υπάρχουν αρκετά διαθέσιμα, εμείς όμως για τον σκοπό μας θα χρησιμοποιήσουμε το Raspbian, με την έκδοση "New Out Of Box Software (NOOBS)". Εφόσον ολοκληρωθεί η διαδικασία εγκατάστασης, το Raspberry Pi είναι εντελώς έτοιμο για χρήση.

### **RASPBIAN CONFIGURATION**

Το πρώτο πράγμα που πρέπει να κάνουμε, είναι να ανοίξουμε ένα terminal και να πληκτρολογήσουμε την παρακάτω εντολή:

- **Sudo raspi-config**

Κατόπιν, επιλέγουμε την εντολή "expand filesystem" και κάνουμε μια επανεκκίνηση του Raspberry Pi. Πιθανότατα θα μας πει ότι χρησιμοποιείται ήδη ο μέγιστος διαθέσιμος χώρος, αλλά καλό θα ήταν να το κάνουμε κι αυτό. Στη συνέχεια θα πρέπει να πληκτρολογήσουμε τις παρακάτω εντολές για να εγκαταστήσουμε στο Raspberry Pi τις τελευταίες ενημερώσεις, καθώς επίσης και μερικά χρήσιμα εργαλεία. Οι εντολές που θα χρησιμοποιηθούν είναι οι εξής:

- **Sudo apt-get update**
- **Sudo apt-get upgrade --yes**
- **sudo apt-get install nano git-core python-dev bison libasound2-dev libportaudio-dev python-pyaudio --yes**
- **sudo apt-get remove python-pip**
- **sudo easy\_install pip**

Αφού τελειώσουμε με αυτό, σειρά έχει να δημιουργήσουμε έναν ALSA configuration file. Ο ALSA είναι το περιβάλλον ελέγχου μίξης και έντασης των

ηχητικών πληροφοριών του συστήματός μας. Τοποθετούμε το μικρόφωνο και πληκτρολογούμε:

- **sudo nano /lib/modprobe.d/jasper.conf**

Προσθέτουμε τις παρακάτω εντολές:

- **# Load USB audio before the internal soundcard**
- **options snd\_usb\_audioindex=0**
- **options snd\_bcm2835 index=1**
  
- **# Make sure the sound cards are ordered the correct way in ALSA**
- **options sndslots=snd\_usb\_audio,snd\_bcm2835**

Προφανώς οτιδήποτε εμπεριέχεται στην γραμμή με τις διαίσεις, είναι επεξηγηματικού χαρακτήρα και δεν παίζει κάποιο ρόλο στον κώδικα. Μετά από αυτά τα βήματα είναι απαραίτητη μια επανεκκίνηση:

- **sudo shutdown -r now, ή**
- **sudo halt**

Έπειτα, πρέπει να ελέγξουμε ότι το μικρόφωνο δουλεύει.

Θα επιχειρήσουμε να κάνουμε μια ηχογράφιση με την παρακάτω εντολή:

- **arecord temp.wav**

Δεν ξεχνάμε να ελέγξουμε ότι τα ηχεία ή τα ακουστικά μας είναι συνδεδεμένα στο audio jack του Raspberry Pi. Μπορούμε να αναπαράγουμε το αρχείο που μόλις ηχογραφήσαμε με την εντολή:

- **aplay -D hw:1,0 temp.wav**

Μετά, πρέπει να προσθέσουμε το παρακάτω κομμάτι κώδικα στο τέλος του αρχείου **~/.bash\_profile**:

(Ίσως χρειαστεί να τρέξουμε την εντολή:

- **touch ~/.bash\_profile** στην περίπτωση που το αρχείο δεν υπάρχει ήδη).



- `export LD_LIBRARY_PATH="/usr/local/lib"`
- `source .bashrc`

Προσθέτουμε επίσης τα παρακάτω στον φάκελο `~/.bashrc` ή στον `~/.bash_profile`:

- `LD_LIBRARY_PATH="/usr/local/lib"`
- `export LD_LIBRARY_PATH`
- `PATH=$PATH:/usr/local/lib/`
- `export PATH`

Μετά από όλα αυτά, είμαστε πλέον έτοιμοι να εγκαταστήσουμε το «Jasper».

### **ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ JASPER**

Ανοίγουμε ένα terminal και εφόσον βρισκόμαστε στο home directory του Pi, κλωνοποιούμε τον πηγαίο κώδικα του Jasper με την παρακάτω εντολή:

- `git clone https://github.com/jasperproject/jasper-client.git jasper`

Το Jasper απαιτεί διάφορες βιβλιοθήκες της Python, τις οποίες μπορούμε να τις εγκαταστήσουμε όλες με τις παρακάτω εντολές:

- `sudo pip install --upgrade setuptools`
- `sudo pip install -r jasper/client/requirements.txt`

Κάποιες φορές ίσως να είναι απαραίτητο να κάνουμε το jasper εκτελέσιμο, και αυτό γίνεται ως εξής:

- `chmod +x jasper/jasper.py`

Τώρα πια έχουμε εγκαταστήσει τον πυρήνα του λογισμικού του Jasper. Σειρά έχει η εγκατάσταση των dependencies.

### **ΕΓΚΑΤΑΣΤΑΣΗ ΤΩΝ DEPENDENCIES**

Για να είναι σε θέση να αντιληφθεί το Jasper τι του λέμε, είναι απαραίτητη μια Μηχανή Ομιλίας-σε-κείμενο (Speech-to-Text / STT Engine) όπως έχει ήδη αναφερθεί. Επιπροσθέτως, το Jasper χρειάζεται μια Μηχανή Κειμένου-σε-ομιλία (Text-to-Speech Engine / TTS Engine) για να μπορεί να απαντάει στις εντολές που του δίνουμε. Σκοπός του συγκεκριμένου Έξυπνου Βοηθού είναι η ευελιξία, μεταξύ άλλων, οπότε μας δίνει την δυνατότητα να επιλέξουμε οποιαδήποτε Μηχανή TTS ή STT θέλουμε εμείς. Ανάλογα με την επιλογή μας φυσικά, ίσως χρειαστεί να εγκαταστήσουμε κάποια επιπλέον προγράμματα. Παρακάτω θα δούμε αναλυτικά ποιές Μηχανές έχουμε στη διάθεσή μας, τι προσφέρουν, και με βάση κάποια κριτήρια, θα μπορούμε να έχουμε μια εικόνα σχετικά με το ποιά είναι η καλύτερη δυνατή επιλογή αναλόγως με το τι θέλει ο χρήστης.

### **DEPENDENCIES ΓΙΑΤΟ POCKETSPHINX STT ENGINE**

Το Jasper μπορεί να χρησιμοποιήσει τη μηχανή «PocketSphinx» ως μέσο αναγνώρισης φωνής. Εάν θέλουμε να χρησιμοποιήσουμε τη συγκεκριμένη μηχανή θα πρέπει να διαθέσουμε αρκετό χρόνο. Σε πρώτη φάση θα πρέπει να εγκαταστήσουμε τα εξής:

- sphinxbase&pocketsphinx
- CMUCLMTK
- MIT LanguageModelingToolkit
- m2m-aligner
- OpenFST&Phonetisaurus

### **ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ DEPENDENCY SPHINXBASE / POCKETSPHINX**

Πρώτα από όλα πρέπει να εγκαταστήσουμε το Pocketsphinx. Επειδή εμείς χρησιμοποιούμε λειτουργικό σύστημα Raspbian θα πρέπει να κάνουμε compile και

εγκατάσταση από την πηγή (source). Αυτό για να επιτευχθεί θα πρέπει να πληκτρολογήσουμε τις παρακάτω εντολές:

- **Wget**<http://downloads.sourceforge.net/project/cmusphinx/sphinxbase/0.8/sphinxbase-0.8.tar.gz>
- **tar -zxvf sphinxbase-0.8.tar.gz**
- **cd ~/sphinxbase-0.8/**
- **./configure --enable-fixed**
- **make**
- **sudo make install**
- **wget**  
**http://downloads.sourceforge.net/project/cmusphinx/pocketsphinx/0.8/pocketsphinx-0.8.tar.gz**
- **tar -zxvf pocketsphinx-0.8.tar.gz**
- **cd ~/pocketsphinx-0.8/**
- **./configure**
- **make**
- **sudo make install**
- **cd ..**
- **sudo easy\_install pocketsphinx**

### **ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ DEPENDENCY “CMUCLMTK”**

Ξεκινάμε εγκαθιστώντας μερικά dependencies:

- **sudo apt-get install subversion autoconf libtool automake gfortran g++ --yes**

Στη συνέχεια, πηγαίνουμε στο home directory, ή στο directory που έχουμε εγκαταστήσει το Jasper για να εγκαταστήσουμε το CMUCLMTK:

- **svn co https://svn.code.sf.net/p/cmusphinx/code/trunk/cmuclmtk/**
- **cd cmuclmtk/**
- **./autogen.sh && make && sudo make install**
- **cd ..**

Εφόσον αυτή η διαδικασία ολοκληρωθεί, φεύγουμε από το directory που βρισκόμαστε και κατεβάζουμε τις ακόλουθες βιβλιοθήκες.

## ΕΓΚΑΤΑΣΤΑΣΗ ΤΩΝ ΒΙΒΛΙΟΘΗΚΩΝ PHONETISAURUS, M2M-

### ALINGERKAIMITLM

Για να χρησιμοποιήσουμε τη μηχανή Pocketsphinx, θα χρειαστεί επίσης να εγκαταστήσουμε το «MIT language Modeling Toolkit», και το «Phonetisaurus» (άρα λοιπόν και το Open FST).

Ξεκινάμε λοιπόν πληκτρολογώντας τις εξής εντολές:

- **wget <http://distfiles.macports.org/openfst/openfst-1.3.4.tar.gz>**
- **wget [https://github.com/mitlm/mitlm/releases/download/v0.4.1/mitlm\\_0.4.1.tar.gz](https://github.com/mitlm/mitlm/releases/download/v0.4.1/mitlm_0.4.1.tar.gz)**
- **wget <https://storage.googleapis.com/google-code-archive-downloads/v2/code.google.com/m2m-aligner/m2m-aligner-1.2.tar.gz>**
- **wget <https://storage.googleapis.com/google-code-archive-downloads/v2/code.google.com/phonetisaurus/is2013-conversion.tgz>**

Εξάγουμε τα αρχεία μας:

- **tar -xvf m2m-aligner-1.2.tar.gz**
- **tar -xvf openfst-1.3.4.tar.gz**
- **tar -xvf is2013-conversion.tgz**
- **tar -xvf mitlm\_0.4.1.tar.gz**

Χτίζουμε το Open FST:

- **cd openfst-1.3.4/**
- **sudo ./configure --enable-compact-fsts --enable-const-fsts --enable-far --enable-lookahead-fsts --enable-pdt**
- **sudomakeinstall**

Χτίζουμε το M2M, το MITLMT και το Phonetisaurus:

- **cdm2m-aligner-1.2/**
- **sudomake**
- **cdmitlm-0.4.1/**

- `sudo ./configure`
- `sudomakeinstall`
- `cdis2013-conversion/phonetisaurus/src`
- `sudomake`

Μετακινούμε κάποια από τα αρχεία (compiled files):

- `sudo cp ~/m2m-aligner-1.2/m2m-aligner /usr/local/bin/m2m-aligner`
- `sudo cp ~/is2013-conversion/bin/phonetisaurus-g2p /usr/local/bin/phonetisaurus-g2p`

Χτίζουμε το Phonetisaurus FST μοντέλο:

- `wget https://www.dropbox.com/s/kfht75czdwucni1/g014b2b.tgz`
- `tar -xvf g014b2b.tgz`

Χτίζουμε το Phonetisaurus μοντέλο:

- `cd g014b2b/`
- `./compile-fst.sh`
- `cd ..`

Τελικώς, επονομάζουμε τον παρακάτω φάκελο για λόγους ευχέρειας:

- `mv ~/g014b2b ~/phonetisaurus`

Όταν ολοκληρωθεί όλη η διαδικασία, τότε κάνουμε μια επανεκκίνηση το Raspberry Pi. Σε αυτό το σημείο έχουμε πραγματοποιήσει την εγκατάσταση του Jasper και όλων των απαραίτητων βοηθητικών λογισμικών για να το τρέξουμε. Προτού ξεκινήσουμε να παίζουμε με αυτό, ωστόσο, πρέπει να διαμορφώσουμε το προφίλ του Jasper και να του παρέχουμε μερικές βασικές πληροφορίες.

Εάν όμως κάποιος θέλει να δοκιμάσει κάτι διαφορετικό από τη μηχανή Rocketsphinx, υπάρχουν και κάποιες άλλες επιλογές. Παρακάτω θα εξετάσουμε ποιες είναι αυτές και τι χρειάζεται να κάνει κανείς για να τις εγκαταστήσει.

### **ΕΓΚΑΤΑΣΤΑΣΗ ΤΩΝ DEPENDENCIES ΓΙΑ ΤΗ ΜΗΧΑΝΗ JULIUS STT**

Ξεκινώντας, πρέπει να κάνουμε compile τη μηχανή Julius, χειροκίνητα:

- `sudo apt-get update`
- `sudo apt-get install build-essential zlib1g-dev flex libasound2-dev libesd0-dev libsndfile1-dev`

Έπειτα, κατεβάζουμε το “Julius source tarball” και το εξάγουμε στο: ~/Julius.

- `cd ~/julius`
- `./configure --enable-words-int`
- `make`
- `sudomakeinstall`

Να σημειωθεί ότι για τη συγκεκριμένη μηχανή, θα χρειαστούμε ένα ακουστικό μοντέλο και ένα λεξικό.

### **ΕΓΚΑΤΑΣΤΑΣΗ ΤΩΝ DEPENDENCIES ΓΙΑ ΤΟ FESTIVAL TTS ENGINE**

Εγκαθιστούμε το festival και το festvox-don:

- `sudo apt-get update`
- `sudo apt-get install festival festvox-don`

### **ΕΓΚΑΤΑΣΤΑΣΗ ΤΩΝ DEPENDENCIES ΓΙΑ ΤΟ FLITE TTS ENGINE**

Εγκαθιστούμε το flite:

- `sudo apt-get update`
- `sudo apt-get install flite`

### **ΕΓΚΑΤΑΣΤΑΣΗ ΤΩΝ DEPENDENCIES ΓΙΑ ΤΟ SVOX Pico TTS ENGINE**

Ομοίως με παραπάνω:

- `sudo apt-get update`
- `sudo apt-get install libttspico-utils`

### **ΕΓΚΑΤΑΣΤΑΣΗ ΤΩΝ DEPENDENCIES ΓΙΑ ΤΟ GOOGLE TTS ENGINE**

Γι' αυτή τη μηχανή θα χρειαστούμε τα python-pymad μέσω APT και το gTTS μέσω PIP:

- **sudo apt-get update**
- **sudo apt-get install python-pymad**
- **sudo pip install --upgrade gTTS**

### **ΕΓΚΑΤΑΣΤΑΣΗ ΤΩΝ DEPENDENCIES ΓΙΑ ΤΟ IVONA TTS ENGINE**

Παρόμοια διαδικασία με το Google TTS μοντέλο, θα χρειαστούμε τα python-pymad μέσω APT και το pyvona μέσω PIP:

- **sudo apt-get update**
- **sudo apt-get install python-pymad**
- **sudo pip install --upgradepyvona**

### **ΔΙΑΜΟΡΦΩΣΗ ΤΟΥ ΠΡΟΦΙΛ ΤΟΥ JASPER**

Ο Jasper χρειάζεται ένα φάκελο διαμόρφωσης που αποκαλείται προφίλ. Για να είναι σε θέση ο Jasper να αναφέρει με ακρίβεια τοπικές καιρικές συνθήκες, να μας στέλνει μηνύματα και πολλά άλλα, πρωτίστως θα πρέπει να δημιουργήσουμε ένα προφίλ χρήστη. Για να διευκολυνθεί η διαδικασία, τρέχουμε το προφίλ πληθυσμού το οποίο έρχεται πακεταρισμένο με το Jasper:

- **cd ~/jasper/client**
- **python populate.py**

Από εκεί και πέρα, πληκτρολογούμε τις απαραίτητες πληροφορίες ή εάν θέλουμε να παρακάμψουμε κάποιες, απλώς πατάμε "enter". Εξ' ορισμού, το προφίλ που θα προκύψει, αποθηκεύεται ως ένα YML αρχείο, στο εξής path: **~/jasper/profile.yml**

Κάτι το οποίο αξίζει να σημειωθεί καθώς έγκειται θέμα ασφαλείας, είναι το παρακάτω: Το script “populate.py” θα ζητήσει τον κωδικό πρόσβασης του Gmail του χρήστη. Φυσικά, αυτό είναι εντελώς προαιρετικό και δεν θα εξαχθεί με κανέναν τρόπο από τη συσκευή, μέσω του Jasper (εάν κάποιος τρίτος προσπαθήσει να αποκτήσει πρόσβαση, είναι άλλη υπόθεση).

Ο κωδικός αυτός λοιπόν, επιτρέπει στο Jasper να μας ενημερώνει σχετικά με e-mails ή να μας στέλνει ειδοποιήσεις μέσω SMS (εάν βάλουμε τον αριθμό του κινητού μας), αλλά το θέμα είναι ότι οι πληροφορίες αυτές θα αποθηκευτούν ακριβώς όπως τις πληκτρολογήσαμε, χωρίς να κρυπτογραφηθούν, στο αρχείο profile.yml.

Ωστόσο, μια εναλλακτική για να αποφύγουμε μια τέτοια διαρροή, είναι να βάλουμε απλώς την ηλεκτρονική μας διεύθυνση χωρίς τον κωδικό πρόσβασής μας, και θα χρησιμοποιηθεί κανονικά για ενημερώσεις κ.λ.π., εάν διαμορφώσουμε ένα λογαριασμό Mailgun, όπως θα δούμε σε λίγο.

### **ΕΠΙΛΟΓΗ ΤΗΣ ΚΑΤΑΛΛΗΛΗΣ Sound-To-Text ΜΗΧΑΝΗΣ (Ομιλίας-Σε-Κείμενο)**

Πρέπει να επιλέξουμε ποιά Μηχανή Ομιλίας-Σε-Κείμενο θα χρησιμοποιεί ο Jasper. Μια τέτοια Μηχανή, είναι στην πραγματικότητα ένα software το οποίο είναι ικανό να μετατρέψει ηχογραφημένη ομιλία σε κείμενο. Σε περίπτωση που ο χρήστης πει κάτι αλλά ο Jasper καταλάβει κάτι διαφορετικό από αυτό που είπε ο χρήστης, τότε είναι είτε πρόβλημα του μικροφώνου, είτε λάθος παραμετροποιημένη η Μηχανή. Άρα λοιπόν, η επιλογή της κατάλληλης Μηχανής Ομιλίας-Σε-Κείμενο έχει καθοριστική σημασία, ούτως ώστε να επιτευχθεί ομαλή λειτουργία του Jasper. Ενώ τα περισσότερα εργαλεία αναγνώρισης ομιλίας βασίζονται μόνο σε μία SST Μηχανή, ο Jasper είναι ευέλικτος και υποστηρίζει διάφορες επιλογές Μηχανών STT.



Ας εξετάσουμε ποιες είναι οι επιλογές μας:

- 1. Pocketsphinx:** Το Pocketsphinx είναι ένα decoder ανοικτού κώδικα, που αναπτύχθηκε από την εταιρεία CMUSphinx. Είναι γρήγορο και σχεδιασμένο να δουλεύει ικανοποιητικά σε ενσωματωμένα συστήματα όπως το Raspberry Pi. Δυστυχώς όμως, το ποσοστό αναγνώρισης δεν είναι το καλύτερο και επίσης έχει πολλά dependencies, άρα χρειάζεται να δαπανηθεί αρκετός χρόνος και αρκετή έρευνα για να επιλυθούν τα ενδεχόμενα σφάλματα που θα συναντήσει ο χρήστης. Από την άλλη όμως, η αναγνώριση πραγματοποιείται εκτός δικτύου, άρα δεν είναι ανάγκη να έχουμε ενεργό διαδίκτυο στο χώρο για να το χρησιμοποιήσουμε. Είναι η πλέον σωστή επιλογή εάν είμαστε προσεκτικοί με τα προσωπικά μας δεδομένα.
- 2. Google STT:** Αυτή η μηχανή, όπως υποδηλώνει και το όνομά της, είναι ανεπτυγμένη από την εταιρεία Google. Σε περίπτωση που ο χρήστης έχει smartphone, τότε είναι πιθανό να είναι ήδη εξοικειωμένος με τη χρήση της, διότι είναι βασικά η ίδια Μηχανή η οποία εκτελεί την αναγνώριση αν πούμε "OK Google". Μπορεί να μεταγράψει έναν συγκεκριμένο αριθμό λέξεων την ημέρα και χρειάζεται μια ενεργή σύνδεση στο Διαδίκτυο.
- 3. AT & T STT:** Αυτό είναι ένα decoder ομιλίας σχεδιασμένο από τη γνωστή εταιρεία τηλεπικοινωνιών AT & T. Όπως και με τη Μηχανή της Google, χρειάζεται μια ενεργή σύνδεση στο Διαδίκτυο.
- 4. Wit.ai STT:** Η συγκεκριμένη μηχανή βασίζεται στις wit.ai cloud υπηρεσίες και χρησιμοποιεί την τεχνική "crowdsourcing" για να εκπαιδεύσει τους αλγόριθμους αναγνώρισης ομιλίας. Όπως είναι αναμενόμενο, δεδομένης της ύπαρξης cloud υπηρεσίας, είναι απαραίτητη μια ενεργή σύνδεση στο Διαδίκτυο.
- 5. Julius:** Τέλος, η συγκεκριμένη, είναι μια υψηλών-επιδόσεων-ανοικτού-κώδικα-αναγνώρισης-ομιλίας Μηχανή. Δεν απαιτεί ενεργή σύνδεση στο Διαδίκτυο. Εάν θέλει κανείς να χρησιμοποιήσει την εν λόγω Μηχανή, θα χρειαστεί να εκπαιδεύσει από το μηδέν το δικό του ακουστικό μοντέλο, πράγμα το οποίο είναι ένα εξαιρετικά περίπλοκο έργο και απαιτεί εξειδικευμένες προγραμματιστικές γνώσεις. Στους

περισσότερους χρήστες θα ταιριάζει καλύτερα μια Μηχανή από αυτές που είδαμε παραπάνω.

Κάτι το οποίο πρέπει να σημειωθεί, είναι ότι εκτός από το PocketSphinx και το Julius, όλες οι παραπάνω STT μηχανές μεταφέρουν τα δεδομένα που λαμβάνουν από το μικρόφωνο μέσω διαδικτύου. Εάν κάποιος χρήστης δε θέλει η Google, το Wit.ai και η AT & T να ακούν τις συνομιλίες του, τότε θα πρέπει να χρησιμοποιήσει κάποια άλλη μηχανή. Μια καλή επιλογή γι' αυτή την περίπτωση, είναι το PocketSphinx, για παράδειγμα. Επίσης, Μηχανές οι οποίες απαιτούν ενεργή σύνδεση στο διαδίκτυο δεν προτείνονται, από την άποψη της ασφάλειας. Όπως και πριν, το PocketSphinx είναι μια πολύ καλή επιλογή.

### **ΔΙΑΜΟΡΦΩΣΗ ΤΗΣ ΜΗΧΑΝΗΣ ROCKETSPHINXSTT**

Αφού έχουμε ολοκληρώσει την διαδικασία εγκατάστασης όπως είδαμε πιο πάνω, εντοπίζουμε το FST model directory αλλά και το Hidden Markov Model directory, και τοποθετούμε το path του καθενός στο αρχείο "profile.yml":

```
stt_engine:sphinx
pocketsphinx:
fst_model:'../phonetisaurus/g014b2b.fst'
hmm_dir:'/usr/share/pocketsphinx/model/hmm/en_US/hub4wsj_sc_8k'
```

### **ΔΙΑΜΟΡΦΩΣΗ ΤΗΣ ΜΗΧΑΝΗΣ Julius STT**

Παρομοίως, ολοκληρώνουμε την διαδικασία εγκατάστασης, αλλά θα χρειαστούμε και κάτι παραπάνω γι' αυτή τη Μηχανή. Αυτό είναι ένα ακουστικό μοντέλο σε HTK μορφή. Αν και το VoxForge προσφέρει ακουστικά ανεξάρτητα μοντέλα, θα πρέπει ο χρήστης να προσαρμόσει το μοντέλο και να το εκπαιδεύει με τη φωνή του, για να έχει καλά ποσοστά αναγνώρισης. Αυτό είναι μια πολύ σύνθετη και εξειδικευμένη διαδικασία και για να επιτευχθεί χρειάζεται ιδιαίτερη μελέτη επάνω στο θέμα.

Επίσης, η Μηχανή αυτή, χρειάζεται ένα αρχείο το οποίο εξυπηρετεί σαν λεξικό που χαρτογραφεί λέξεις σε φωνήματα και βέβαια πρέπει να περιέχει όλες τις λέξεις που το Julius θα πρέπει να αναγνωρίζει. Ένα πολύ περιεκτικό λεξικό για την περίπτωση αυτή, είναι το VoxForge Lexicon.

Αφού λοιπόν δημιουργήσει ο χρήστης το δικό του ακουστικό μοντέλο, θα πρέπει να προσδιορίσει τα paths για το “hmmdefs”, “tiedlist” και το αρχείο του λεξικού στο αρχείο “profile.yml”:

```
stt_engine:julius
julius:
hmmdefs:'/path/to/your/hmmdefs'
tiedlist:'/path/to/your/tiedlist'
lexicon:'/path/to/your/lexicon.tgz'
lexicon_archive_member:'VoxForge/VoxForgeDict'
```

Να σημειωθεί ότι η τελευταία γραμμή είναι απαραίτητη μόνο στην περίπτωση που το λεξικό είναι σε μορφή “tar” ή “tar.gz”. Τέλος, εάν συναντήσει κανείς σφάλματα ή προειδοποιήσεις του τύπου «voca\_load\_htkdict: line 19: triphone “r-d+v” notfound», ή κάτι αντίστοιχο, σημαίνει ότι προσπαθεί να αναγνωρίσει λέξεις οι οποίες δεν συμπεριλαμβάνονται στο ακουστικό μοντέλο. Για να μπορεί να αναγνωρίσει το Julius τις λέξεις αυτές πρέπει ο χρήστης να εκπαιδεύσει το ακουστικό του μοντέλο.

### **ΔΙΑΜΟΡΦΩΣΗ ΤΗΣ ΜΗΧΑΝΗΣ STT GOOGLE**

Για τη μηχανή αυτή, είναι απαραίτητο ένα «κλειδί» Google Speech API. Για να το αποκτήσει κανείς το «κλειδί» αυτό, θα πρέπει να γίνει μέλος του Chromium Developers Group και να δημιουργήσει ένα project μέσω της «Google Developers Console». Δεν είναι ιδιαίτερα δύσκολη διαδικασία. Εφόσον αυτό επιτευχθεί, βρίσκουμε τον αριθμητικό κωδικό μας στο πεδίο “Credentials”. Στην ουσία, η Google μας φτιάχνει έναν randomized αριθμό (το «κλειδί» που είπαμε παραπάνω), για δημόσια API πρόσβαση. Εφόσον ολοκληρωθεί η διαδικασία λοιπόν, περνάμε τα δεδομένα στο αρχείο “profile.yml”:

**stt\_engine:google**

**keys:**

**GOOGLE\_SPEECH:'your\_google\_speech\_api\_key\_here\_xyz1234'**

### **ΔΙΑΜΟΡΦΩΣΗ ΤΗΣ ΜΗΧΑΝΗΣ AT & T STT**

Η Μηχανή αυτή απαιτεί ένα AT & T app\_key και ένα app\_secret κωδικό, να εμπεριέχεται στο αρχείο “profile.yml”. Για να γίνει αυτό, πρέπει ο χρήστης να κάνει εγγραφή στο επίσημο site και να δημιουργήσει ένα καινούργιο app. Από εκεί και πέρα το περνάει στο αρχείο “profile.yml” και είναι έτοιμο:

**stt\_engine:att**

**att-stt:**

**app\_key:app\_key\_here**

**app\_secret:app\_secret\_here**

### **ΔΙΑΜΟΡΦΩΣΗ ΤΗΣ ΜΗΧΑΝΗΣ Wit.ai STT**

Αυτή η εφαρμογή απαιτεί ένα Wit.ai Access Token να εμπεριέχεται στο αρχείο “profile.yml”. Για να αποκτήσει κανείς ένα τέτοιο token, πρέπει να κάνει εγγραφή στο site της Wit.ai και να αντιγράψει το token στο αρχείο “profile.yml”:

**stt\_engine:witai**

**witai-stt:**

**access\_token:access\_token\_here**

### **ΕΠΙΛΟΓΗ TTS ΜΗΧΑΝΗΣ**

Μια TTS Μηχανή (Text-To-Sound Engine), κάνει το ακριβώς αντίθετο από μια STT Μηχανή: παίρνει γραμμένο κείμενο και το μετατρέπει σε ομιλία. Το Jasper

υποστηρίζει διάφορες TTS μηχανές οι οποίες έχουν διαφορετικά χαρακτηριστικά. Ας δούμε κάποιες από αυτές:

1. **eSpeak**: Πρόκειται για ένα synthesizer ομιλίας ανοικτού κώδικα για αρκετές πλατφόρμες. Η σύνθεση ομιλίας γίνεται εκτός δικτύου, αλλά οι περισσότερες φωνές ακούγονται αρκετά «ρομποτικές».
2. **Festival**: Χρησιμοποιεί το «Festival Speech Synthesis System», ένα synthesizer ομιλίας ανοικτού κώδικα που αναπτύχθηκε από το Center for Speech Technology Research στο Πανεπιστήμιο του Εδιμβούργου. Όπως και το eSpeak, πραγματοποιεί τη σύνθεση ομιλίας εκτός δικτύου.
3. **Flite**: Χρησιμοποιεί το “CMUFlite”, μια ελαφριά και γρήγορη μηχανή σύνθεσης ομιλίας, η οποία αρχικά σχεδιάστηκε για μικρές ενσωματωμένες μηχανές. Συνθέτει και αυτό την ομιλία εκτός δικτύου.
4. **SVOX Pico TTS**: Αυτή ήταν η Μηχανή Κειμένου-σε-ομιλία που χρησιμοποιήθηκε στο Android 1.6 “Donut”. Είναι μια ανοικτού κώδικα εφαρμογή, η οποία λειτουργεί και αυτή εκτός δικτύου. Η ποιότητα είναι καλύτερη από το eSpeak και το Festival.
5. **Google TTS**: Με αυτή τη Μηχανή, η σύνθεση ομιλίας γίνεται στους servers της Google, πράγμα το οποίο σημαίνει ότι χρειάζεται σύνδεση στο διαδίκτυο για να δουλέψει, άρα από την άποψη της ιδιωτικότητας και της ασφάλειας δεν είναι καλή επιλογή. Κατά τα άλλα όμως, είναι μια εξαιρετική επιλογή.
6. **Ivona TTS**: Χρησιμοποιεί την υπηρεσία της Amazon, Ivona Speech Cloud Service. Η σύνθεση ομιλίας γίνεται με τη χρήση δικτύου, κάτι το οποίο σημαίνει ότι η Amazon μπορεί να ακούσει (εάν θέλει), οτιδήποτε λέει ο χρήστης στο Jasper.
7. **Mary TTS**: Αυτή η Μηχανή είναι ανοικτού κώδικα, γραμμένη στη γλώσσα της Java. Για να την χρησιμοποιήσει κανείς, πρέπει να στήσει τον δικό του server. Επειδή μπορεί ο server να βρίσκεται στο ίδιο μηχάνημα με το Jasper, δεν είναι απαραίτητη η σύνδεση στο διαδίκτυο.

8. **Mac OSX TTS:** Αυτή η Μηχανή δουλεύει μόνο εάν έχουμε στήσει το Jasper σε λειτουργικό Macintosh. Χρησιμοποιεί την εντολή «say» για να συνθέσει την ομιλία.

Εάν χρησιμοποιεί κανείς την Google TTS, ή την Mary TTS με τον server να βρίσκεται σε διαφορετικό υπολογιστή, ό,τιδήποτε λέει το Jasper θα μεταφέρεται μέσω διαδικτύου. Εάν ο χρήστης δε θέλει η Google ή, ενδεχομένως, κάποιος άλλος να ακούει το τι λέει το Jasper, θα πρέπει να μην χρησιμοποιήσει τις Μηχανές αυτές.

### **ΔΙΑΜΟΡΦΩΣΗ ΤΗΣ TTS ΜΗΧΑΝΗΣ eSpeak**

Αφού ολοκληρώσουμε την εγκατάσταση, όπως έχουμε δει αναλυτικά πιο πάνω, επιλέγουμε τη συγκεκριμένη Μηχανή στο αρχείο “profile.yml”:

**Tts\_engine: espeak-tts**

Μπορούμε να προσαρμόσουμε ακόμη περισσότερο τη Μηχανή αυτή, με το να αλλάξουμε την χροιά της φωνής που ακούμε, την τονικότητά της αλλά και το πόσο γρήγορα θα αναπαράγει τις λέξεις. Όλες αυτές οι παράμετροι ρυθμίζονται στο αρχείο “profile.yml”. Ένα παράδειγμα:

**espeak-tts:**

**voice:'default+m3'**

**pitch\_adjustment:40**

**words\_per\_minute:160**

### **ΔΙΑΜΟΡΦΩΣΗ ΤΗΣ TTS ΜΗΧΑΝΗΣ Festival**

Αφού εγκαταστήσουμε τη Μηχανή Festival, απλώς επιλέγουμε τη Μηχανή αυτή στο αρχείο “profile.yml”:

**tts\_engine: festival-tts**

### **ΔΙΑΜΟΡΦΩΣΗ ΤΗΣ TTS ΜΗΧΑΝΗΣ Flite**

Ίδια μεθοδολογία. Εγκατάσταση και επιλογή της Μηχανής στο αρχείο “profile.yml”:

**Tts\_engine: flite-tts**

Εάν θέλουμε να χρησιμοποιήσουμε μια διαφορετική φωνή, το μόνο που έχουμε να κάνουμε είναι να το ορίσουμε:

**Flite-tts:**

**Voice: 'slt'**

Για να δούμε πόσες διαθέσιμες φωνές έχουμε, πληκτρολογούμε την εντολή flite -lv στο command line.

### **ΔΙΑΜΟΡΦΩΣΗ ΤΗΣ TTS ΜΗΧΑΝΗΣ SVOX Pico ΚΑΙ ΤΗΣ ΜΗΧΑΝΗΣ TTS**

#### **GOOGLE**

Ίδια μεθοδολογία. Εγκατάσταση και επιλογή της Μηχανής στο αρχείο “profile.yml”:

**tts\_engine: pico-tts**

**tts\_engine: google-tts**

### **ΔΙΑΜΟΡΦΩΣΗ ΤΗΣ TTS ΜΗΧΑΝΗΣ IVONA**

Εγκαθιστούμε τα απαραίτητα dependencies για να αποκτήσουμε πρόσβαση στην Ivona speech cloud υπηρεσία της Amazon. Θα χρειαστεί να κάνουμε εγγραφή για να χρησιμοποιήσουμε την υπηρεσία αυτή. Τέλος θέτουμε στο αρχείο “profile.yml” τη μηχανή αυτή, όπως επίσης και τα cloud keys. Για παράδειγμα:

**tts\_engine:ivona-tts**

**ivona-tts:**

**access\_key:'access\_key'# required**

**secret\_key:'secret\_key'# required**

```
voice:'Eric'  
region:'eu-west'  
speech_rate:'medium'  
sentence_break:400
```

### ΔΙΑΜΟΡΦΩΣΗ ΤΗΣ TTS ΜΗΧΑΝΗΣ MARY

Απλώς θέτουμε την Μηχανή αυτή στο αρχείο “profile.yml”. Επίσης εάν θέλουμε μπορούμε να αλλάξουμε τον εξ’ ορισμού server με έναν δικό μας:

```
tts_engine:mary-tts  
mary-tts:  
server: 'server_parameters_here'  
port:'59125'  
language:'en_GB'  
voice:'dfki-spike'
```

Τέλος, για το περιβάλλον Macintosh κάνουμε το ίδιο. Τροποποιούμε το αρχείο “profile.yml” ως εξής:

```
tts_engine: osx-tts
```

### ΣΥΓΚΡΙΤΙΚΟΣ ΠΙΝΑΚΑΣ STT ΜΗΧΑΝΩΝ

Όνομα Μηχανής	Διαθέσιμο προς ανάγνωση και τροποποίηση	Αναγκαιότητα Δικτύου	Επιτυχές Ποσοστό Αναγνώρισης
PocketSphinx	Ανοικτού κώδικα	Μη απαραίτητη	Κάτω του μετρίου
Google	Ιδιόκτητος κώδικας	Απαραίτητη	Πολύ καλό, αλλά περιορισμένο
AT & T	Ιδιόκτητος κώδικας	Απαραίτητη	Μέτριο
Wit.ai	Ιδιόκτητος κώδικας	Απαραίτητη	Μέτριο
Julius	Ανοικτού κώδικα	Μη απαραίτητη	Εξαρτάται από την εκπαίδευση του μοντέλου



**ΣΥΓΚΡΙΤΙΚΟΣ ΠΙΝΑΚΑΣ ΤΤΣ ΜΗΧΑΝΩΝ**

Όνομα Μηχανής	Διαθέσιμο προς ανάγνωση και τροποποίηση	Αναγκαιότητα Δικτύου	Ρύθμιση Φωνής
eSpeak	Ανοικτού κώδικα	Μη απαραίτητη	Εφικτή
Festival	Ανοικτού κώδικα	Μη απαραίτητη	Περιορισμένη
Flite	Ιδιόκτητος κώδικας	Μη απαραίτητη	Περιορισμένη
SVOX Pico	Ανοικτού κώδικα	Μη απαραίτητη	Εφικτή
Google	Ιδιόκτητος κώδικας	Απαραίτητη	Περιορισμένη
Ivona	Ιδιόκτητος κώδικας	Απαραίτητη	Περιορισμένη
Mary	Ανοικτού κώδικα	Αναλόγως τη θέση του Server	Εφικτή

**ΛΟΙΠΕΣ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗΜΕΣ ΕΠΙΛΟΓΕΣ**

**ΧΡΗΣΗ ΤΗΣ ΥΠΗΡΕΣΙΑΣ MAILGUN ΣΑΝ ΕΝΑΛΛΑΚΤΙΚΗ ΤΟΥ GMAIL**

Στην περίπτωση που κάποιος δε θα ήθελε να βάλει τον κωδικό του gmail λογαριασμού του, μπορεί να δημιουργήσει ένα δωρεάν Mailgun λογαριασμό που θα χρησιμοποιεί το Jasper για να στέλνει ειδοποιήσεις. Είναι πολύ εύκολη διαδικασία και το Jasper είναι έτοιμο εξ' ορισμού για άμεση αλληλεπίδραση με την υπηρεσία αυτή. Ωστόσο, το να βάλει κανείς την ηλεκτρονική διεύθυνση είναι ακόμη απαραίτητο, διότι χωρίς αυτή, το Jasper θα μπορεί να στέλνει ειδοποιήσεις μόνο μέσω γραπτού μηνύματος. Όταν ολοκληρώσουμε τη διαδικασία, ορίζουμε την υπηρεσία στο αρχείο "profile.yml»:

**Mailgun:**

username: [usernamehere@sandbox12345.mailgun.org](mailto:usernamehere@sandbox12345.mailgun.org)

password: your\_password\_here

### ΠΡΟΒΛΕΨΗ ΚΑΙΡΟΥ ΠΕΡΑΝ ΤΩΝ ΗΝΩΜΕΝΩΝ ΠΟΛΙΤΕΙΩΝ ΑΜΕΡΙΚΗΣ

Εάν θέλει κανείς να χρησιμοποιήσει το module για την πρόβλεψη του καιρού, αλλάζει εκτός Η.Π.Α., μπορεί να βρει τον αριθμό του τοπικού μετεωρολογικού σταθμού κοντά στο μέρος στο οποίο ζει, μέσω του wunderground.com. Υπάρχει ένας μοναδικός αριθμός (WMOID) ο οποίος χαρακτηρίζει μετεωρολογικούς σταθμούς ανά τον κόσμο και έχει εκδοθεί από τον Παγκόσμιο Μετεωρολογικό Οργανισμό (WMO). Εφόσον βρει το σταθμό που είναι κοντά στο μέρος στο οποίο ζει ο χρήστης, τον τοποθετεί στο αρχείο "profile.yml":

```
wmo_id: 12345
```

### ΕΝΣΩΜΑΤΩΣΗ ΤΟΥ SPOTIFY

Το Jasper έχει την ικανότητα να αναπαράγει playlists από τις βιβλιοθήκες του Spotify. Αυτό προαπαιτεί έναν Premium λογαριασμό. Για να διαμορφώσουμε το Spotify στο Jasper πρέπει να ακολουθήσουμε την παρακάτω διαδικασία:

Εγκαθιστούμε το Mopidy ως εξής:

- **wget -q -O - http://apt.mopidy.com/mopidy.gpg | sudo apt-key add -**
- **sudo wget -q -O /etc/apt/sources.list.d/mopidy.list http://apt.mopidy.com/mopidy.list**
- **sudo apt-get update**
- **sudo apt-get install mopidymopidy-spotify -yes**

Πρέπει να ενεργοποιήσουμε την IPv6:

- **sudo modprobe ipv6**
- **echo ipv6 | sudo tee -a /etc/modules**

Δημιουργούμε ένα αρχείο με όνομα .asoundrc στο παρακάτω path:

```
sudo nano /root/.asoundrc
```

και εισάγουμε τις παρακάτω πληροφορίες:

```
pcm.!default{
    type hw
        card 1
}
ctl.!default{
    type hw
        card 1
}
```

Έπειτα πρέπει να δημιουργήσουμε τους παρακάτω φακέλους και να διαγράψουμε το εξ' ορισμού startup script:

- `sudo mkdir /root/.config`
- `sudo mkdir /root/.config/mopidy`
- `sudo rm /etc/init.d/mopidy`

Τώρα πληκτρολογούμε την εντολή:

```
sudo nano /root/.config/mopidy/mopidy.conf
```

και εισάγουμε τα παρακάτω:

```
[spotify]
username = YOUR_SPOTIFY_USERNAME
password = YOUR_SPOTIFY_PASSWORD
```

```
[mpd]
hostname = ::
```

```
[local]
media_dir = ~/music
```

```
[scrobbler]
enabled = false
```

[audio]

output =alsasink

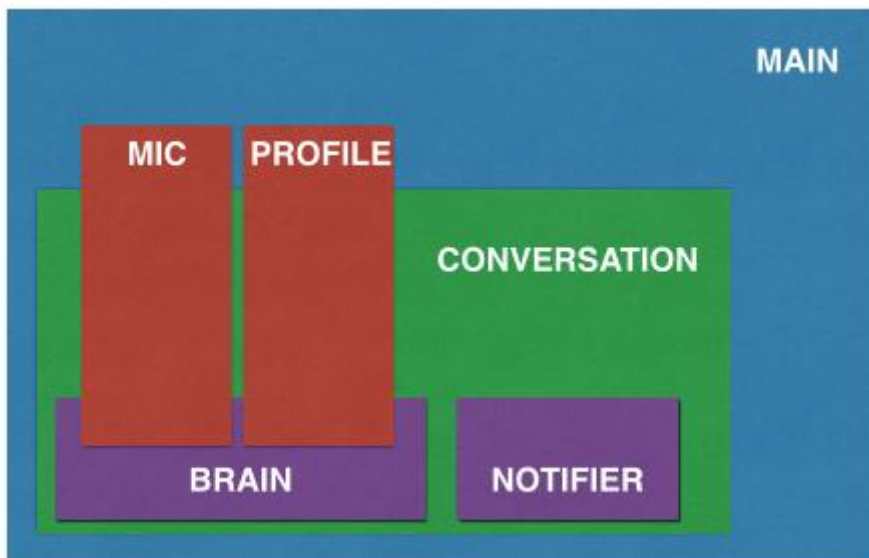
Τέλος, τροποποιούμε το **crontab** για να τρέχει το Moridy. Πληκτρολογούμε την εντολή **sudo crontab -e** και εισάγουμε το εξής:

**@reboot moridy;**

Με την επανεκκίνηση του Jasper, αν η εγκατάσταση και η παραμετροποίηση έχει γίνει σωστά, θα πρέπει να βλέπουμε μια “spotify” εντολή η οποία μας παραπέμπει στο “spotify mode”.

### ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΛΟΓΙΣΜΙΚΟΥ

Έχοντας εγκαταστήσει και διαμορφώσει το Jasper και τις απαιτούμενες βιβλιοθήκες, μπορούμε τώρα να ρίξουμε μια ματιά στο πως αλληλεπιδρούν και στο ποια είναι η δομή του κώδικα. Το πρόγραμμα είναι φτιαγμένο από διάφορα «εξαρτήματα».



Εικόνα 5: Αρχιτεκτονική του Λογισμικού Jasper

Το “jasper.py” είναι το βασικό πρόγραμμα το οποίο ορχηστρώνει όλο το Jasper. Δημιουργεί παραδείγματα αλληλεπίδρασης με το “mic”, το “profile” και το “converstation”. Έπειτα, το παράδειγμα του “conversation” «ταΐζει» το “mic” και το “profile” σαν inputs, από τα οποία δημιουργεί ένα “brain” και ένα “notifier”. Το “brain” στη συνέχεια λαμβάνει τα δεδομένα από το “mic” και το “profile” τα οποία βρίσκονται στο “main” και φορτώνει όλα τα διαδραστικά δεδομένα στη μνήμη. Το “brain” είναι ουσιαστικά η διεπαφή μεταξύ των γραμμένων modules (scripts δηλαδή), και ολόκληρου του βασικού πλαισίου. Κάθε module πρέπει να εμπεριέχει τα ορίσματα “isValid()” και “handle()” όπως επίσης και το όρισμα “WORDS = [ . . . ]”. Παρακάτω θα δούμε περισσότερα σχετικά με το πως μπορεί να γράψει κανείς το δικό του κομμάτι κώδικα για το Jasper. Εφόσον λοιπόν ολοκληρώθηκε η διαδικασία που πρέπει να ακολουθηθεί για να δουλέψει το jasper, μπορούμε να ξεκινήσουμε να το χρησιμοποιούμε.

### **EKKINHSEH TOY JASPER**

Για να πραγματοποιηθεί εκκίνηση του προγράμματος, ανοίγουμε ένα παράθυρο τερματικού στο raspberry pi μας, και πληκτρολογούμε την εξής εντολή:

- **`/home/pi/jasper/jasper.py`**

Επίσης, εάν θέλουμε μπορούμε να ορίσουμε το Jasper να ξεκινάει με κάθε boot του raspberry pi. Για να γίνει αυτό θα πρέπει να πληκτρολογήσουμε τις εξής εντολές:

- **`Crontab -e`**
- **`@reboot /home/pi/jasper/jasper.py`**

Φυσικά αυτό είναι το προεπιλεγμένο μονοπάτι το οποίο διαλέγει εξ’ ορισμού το Jasper κατά τη διαδικασία εγκατάστασης. Ο χρήστης μπορεί να κάνει την εγκατάσταση όπου αυτός επιθυμεί.

### ΑΛΛΗΛΕΠΙΔΡΑΣΗ ΧΡΗΣΤΗ & JASPER

Η λογική για να χρησιμοποιήσει κανείς το Jasper είναι η εξής:

- *Χρήστης: «Jasper?»*
- *Jasper: «δυνατό beer»*
- *Χρήστης: «ερώτηση σχετική με κάποιο module»*
- *Jasper: «χαμηλό beer»*
- *Jasper: «απάντηση στην ερώτηση του χρήστη»*

Αφού ο χρήστης πει τη λέξη “Jasper”, θα πρέπει να περιμένει έως ότου ακουστεί το πρώτο beer, που είναι σχετικά δυνατό. Αφού γίνει αυτό, υπάρχει ένα μικρό χρονικό περιθώριο για να ρωτήσει αυτό που επιθυμεί. Αν ξεπεραστεί αυτό το χρονικό διάστημα, ο Jasper θα σταματήσει να ακούει και θα πρέπει να επαναληφθεί η διαδικασία από την αρχή. Κάποια παραδείγματα μπορούμε να δούμε εδώ:

- *Ερώτηση σχετικά με την ώρα: “What’s the time?”*
- *Ερώτηση σχετικά με τον καιρό: “How’s the weather? What’s the weather tomorrow?”*
- *Ερώτηση σχετικά με τα νέα: “What’s in the news?”*
- *Ερώτηση σχετικά με τα e-mails: “Do I have any e-mails?”*
- *Ερώτηση σχετικά με την ιστοσελίδα «hacker news»: “What’s on hacker news?”*
- *Ερώτηση σχετικά με τις ειδοποιήσεις του Facebook: “Facebook notifications?”*
- *Ερώτηση σχετικά με τις εορτές: “Who has a birthday today?”*
- *Ανέκδοτο: “tell me a joke?”*
- *Ερώτηση σχετικά με τη ζωή: “what’s the meaning of life?”*

Παρακάτω θα δούμε με ποιον τρόπο μπορεί κανείς να φτιάξει το δικό του κομμάτι κώδικα για το Jasper, δηλαδή αυτό που αποκαλούμε “module” ή “script”.

### **SPOTIFY MODE**

Για να αποκτήσουμε πρόσβαση στο Spotify πρέπει να πούμε τη λέξη “Music” ή “Spotify” στο Jasper. Ο τρόπος που λειτουργεί είναι ακριβώς ο ίδιος με όλα τα υπόλοιπα modules.

- *“Play rock playlist”*: Ο Jasper ψάχνει την λίστα με όνομα “rock” στη συλλογή του profile μας στο spotify.
- *“Play”*: Πραγματοποιεί αναπαραγωγή του τραγουδιού.
- *“Pause”*: Πραγματοποιεί παύση του τραγουδιού.
- *“Stop”*: Πραγματοποιεί διακοπή του τραγουδιού.
- *“Louder”*: Αυξάνει την ένταση.
- *“Softer”*: Μειώνει την ένταση.
- *“Next”*: Μεταβαίνει στο επόμενο τραγούδι.
- *“Previous”*: Μεταβαίνει στο προηγούμενο τραγούδι.
- *“Close”*: Κλείνει το spotify.

Στην περίπτωση που ο χρήστης αντιμετωπίζει δυσκολία στην αναγνώριση της φωνής του από το Jasper, μπορεί να οφείλεται στο ότι η μουσική είναι πολύ δυνατά. Μια καλή λύση είναι να χαμηλώσει την ένταση της μουσικής, να μιλήσει πιο κοντά στο μικρόφωνο ή να απομακρύνει το μικρόφωνο από τα ηχεία.

### **ΠΩΣ ΔΗΜΙΟΥΡΓΕΙΤΑΙ ΕΝΑ ΑΠΛΟ MODULE ΓΙΑ ΤΟ JASPER**

Με την λέξη module, εννοούμε ένα αρχείο κώδικα, προγραμματισμένο να εκτελεί όποιες ενέργειες του έχουν οριστεί κατά τη δημιουργία του. Στο Jasper, είναι σχετικά εύκολο να προγραμματίσει κανείς ένα module, το μόνο που πρέπει να κατανοήσει, είναι κάποια βασικά σημεία.

Το πρώτο σημείο είναι αυτό που βρίσκεται στην αρχή του κώδικα, και είναι το όρισμα “WORDS = [...]”. Αυτό το σημείο, εννοεί πως όποιες λέξεις

πληκτρολογηθούν μέσα σε αυτό το διάστημα, αντιστοιχούν στο συγκεκριμένο module. Για παράδειγμα, έστω ότι υπάρχει η λέξη “time”. Όταν ο χρήστης προσφωνήσει την λέξη time και αυτή γίνει αντιληπτή από το Jasper, τότε αυτόματα ο έξυπνος βοηθός καταλαβαίνει ότι ο χρήστης ζητάει πληροφορίες για την ώρα. Εφόσον η λέξη έχει ορισθεί και ΔΕΝ υπάρχει κάποιο άλλο module με την λέξη “time” στο πεδίο “WORDS”, τότε εκτελείται κανονικά η ροή του module. Στην περίπτωση που βρίσκονται δύο ίδιες λέξεις σε διαφορετικά modules, η εκτέλεση δεν θα πραγματοποιηθεί και ο χρήστης θα το αντιληφθεί λόγω του ότι το Jasper θα τυπώσει στην οθόνη ένα error. Η λύση στο πρόβλημα αυτό είναι το να ορισθεί μια προτεραιότητα, η οποία γίνεται πολύ απλά με το να πληκτρολογήσει κανείς στο module του, ακριβώς κάτω από το όρισμα “WORDS = [...]”, την λέξη “PRIORITY = 1 , 2 ....”. Φυσικά, μπορεί κανείς να βάλει όποιον αριθμό επιθυμεί, ανάλογα με το τι ανάγκες έχει και το τι προσπαθεί να πετύχει. Με λίγα λόγια, το πρώτο κομμάτι έχει να κάνει με το ποια ή ποιες λέξεις θέλουμε να αναγνωρίζει το Jasper.

Στη συνέχεια πρέπει να οριστεί μια “έγκυρη εισαγωγή (Valid Input)”, η οποία επιστρέφει “True” αν η εισαγωγή είναι έγκυρη για το συγκεκριμένο module, ή “False” αν δεν είναι έγκυρη. Έχει έναν συγκεκριμένο τρόπο σύνταξης, ο οποίος μοιάζει κάπως έτσι:

```
def isValid(text):
```

```
    return bool(re.search(r'\bwhat is the time\b', text, re.IGNORECASE))
```

Στο σημείο που αναγράφονται οι λέξεις “what is the time”, είναι το σημείο που γίνεται ο έλεγχος για το αν θα συνεχιστεί η εκτέλεση του προγράμματος, ή όχι (το true ή false που αναφέραμε προηγουμένως). Αν οι λέξεις οι οποίες αναγράφονται στο σημείο αυτό, υπάρχουν στο αρχικό όρισμα “WORDS”, τότε η εκτέλεση θα συνεχιστεί κανονικά. Σε αντίθετη περίπτωση, η ροή σταματάει και ο χρήστης βλέπει στην οθόνη του κάποιο error (τα errors μπορεί να ποικίλουν, αναλόγως με το που έχει γίνει το λάθος).



Έπειτα, φτάνουμε στο σημείο όπου πρέπει στον κώδικα να ορίσουμε την αναλυθείσα ομιλία, το αντικείμενο μικροφώνου το οποίο χρησιμοποιείται για ομιλία και για να λαμβάνονται επιπλέον εισαγωγές, και τέλος το προφίλ, το οποίο είναι το προφίλ μας, το οποίο στην πραγματικότητα δεν είναι τίποτα περισσότερο από ένα τρόπον τινά «λεξικό», μέσα στο οποίο εμπεριέχονται κάποιες πληροφορίες (μιλήσαμε γι' αυτό παραπάνω).

Ο τρόπος που γίνεται αυτό σε ένα Jasper module, είναι με το όρισμα: `handle(input, mic, profile)`. Ένα παράδειγμα τρόπου σύνταξης, και μάλιστα για το παράδειγμα με την ερώτηση "Jasper , what's the time?" βλέπουμε παρακάτω:

```
def handle(text, mic, profile):  
  
    tz = getTimezone(profile)  
  
    now = datetime.datetime.now(tz=tz)  
  
    service = DateService()  
  
    response = service.convertTime(now)  
  
    mic.say("It is %s right now." % response)
```

Στην τελευταία γραμμή κώδικα, παρατηρούμε πως αναγράφεται η εντολή "mic.say". Ας εξηγήσουμε λοιπόν περί τίνος πρόκειται:

Η εντολή αυτή έχει να κάνει με την αλληλεπίδραση μεταξύ χρήστη και προγράμματος. Συγκεκριμένα, υπάρχουν δύο μέθοδοι που μπορεί να χρησιμοποιήσει κανείς. Η πρώτη είναι με την σύνταξη: `mic.say(message)` η οποία προσφωνεί ότι υπάρχει μέσα στο πεδίο "message", και η δεύτερη σύνταξη, η οποία είναι η εξής: `mic.activeListen()`, κατά την οποία (όταν και εάν χρησιμοποιηθεί), ακούμε ένα beep, πράμα το οποίο σημαίνει πως το Jasper περιμένει να του δώσουμε μια εισαγωγή (ένα input δηλαδή), και στη συνέχεια ακούει έως ότου ο χρήστης σταματήσει την ομιλία. Αυτά είναι τα βασικά σημεία στην δημιουργία ενός module στο Jasper.

## ΚΕΦΑΛΑΙΟ ΤΕΤΑΡΤΟ

### 4. ΣΥΜΠΕΡΑΣΜΑΤΑ – ΑΝΑΦΟΡΕΣ

#### 4.1. Συμπεράσματα Εργασίας

Σκοπός αυτής της ανάλυσης ήταν η διερεύνηση των δυνατοτήτων μερικών από τους ισχύοντες Ε.Π.Β. που υπάρχουν στην αγορά σήμερα. Συμπερασματικά, φαίνεται ότι ο περισσότερο ευέλικτος βοηθός είναι το Cortana το οποίο είναι διαθέσιμο για όλα τα δημοφιλή λειτουργικά συστήματα για κινητά, συμπεριλαμβανομένων και των iOS, Android και Windows 10. Δεν είναι δυνατή η μέτρηση πληροφοριών και τεχνολογιών προς το παρών, κυρίως λόγω του γεγονότος ότι οι Ε.Π.Β. που υπόκεινται σε συζήτηση βρίσκονται υπό κλειστή ανάπτυξη. Ο πλέον ευέλικτος βοηθός λοιπόν, το Cortana, το οποίο επιτρέπει στους προγραμματιστές τρίτων (3<sup>rd</sup> Party) να προσαρμόσουν και να δημιουργήσουν τις δικές τους ενέργειες για τους δικούς τους Ε.Π.Β.

Με την ολοκλήρωση αυτής της μελέτης και αναλυτικής επεξήγησης εγκατάστασης του Έξυπνου Βοηθού, προέκυψαν ενδιαφέροντα συμπεράσματα.

Από την βιβλιογραφική μελέτη προκύπτει ότι οι συγκεκριμένες λύσεις όσων αφορά στο μέγεθος, τις δυνατότητες αλλά και το τελικό συνολικό κόστος σχετικά με τους μικροϋπολογιστές, μπορούν να βρουν άμεση εφαρμογή στην καθημερινή ζωή των ανθρώπων και να εξυπηρετήσουν με διάφορους τρόπους, αναλόγως τις ανάγκες οι οποίες καλούνται να καλυφθούν.

Γενικά, το να χρησιμοποιείται η ομιλία σαν μέσο επικοινωνίας μεταξύ ανθρώπου – υπολογιστή, είναι μια πολύ φιλική και άμεση λύση.

Επίσης μέσω της μελέτης και του σεναρίου εφαρμογής του Έξυπνου Βοηθού Jasper, αναδείχθηκε η πληθώρα υπηρεσιών που μπορεί να παρέχει η συγκεκριμένη επιλογή, σε οποιοδήποτε περιβάλλον.

#### **4.2. Προτάσεις**

Έχει διαπιστωθεί ότι οι Ε.Π.Β. είναι οντότητες με πολλά σύνθετα στοιχεία, τα οποία απαιτούν δεξιότητες στον προγραμματισμό, στα στατιστικά στοιχεία, στη μηχανική μάθηση, στην τεχνητή νοημοσύνη αλλά και στην περιοχή της ηθικής και δεοντολογίας. Η μελλοντική έρευνα στον τομέα αυτό θα μπορούσε να εξελιχθεί βαθύτερα στα εξαρτήματα τα οποία είναι απαραίτητα και ουσιαστικά για την ανάπτυξη και δημιουργία Ε.Π.Β., για τη βιωσιμότητα μεταξύ των εξαρτημάτων, καθώς και την οικοδόμηση μιας οντότητας που επιτρέπει την περαιτέρω επέκταση, από την άποψη των δεξιοτήτων και της απαιτούμενης ικανότητας αλλά και δεοντολογίας που θα έπρεπε να ακολουθηθεί.

## ΠΑΡΑΡΤΗΜΑ

1. *6 December 2016, Jasper Installer*  
[https://github.com/GeoBeBee/mySystem/blob/master/Jasper-dev\\_onPocketPhinx\\_5preAlpha.txt](https://github.com/GeoBeBee/mySystem/blob/master/Jasper-dev_onPocketPhinx_5preAlpha.txt)
2. *Errors during Jasper installation*  
<https://github.com/jasperproject/jasper-client/issues>
3. *Jasper project documentation*  
<http://jasperproject.github.io/documentation/>
4. *13 November 2017, Errors during Jasper installation*  
<https://github.com/jasperproject/jasper-client/issues/648>
5. *5 October 2014, Errors during Jasper installation*  
<https://github.com/jasperproject/jasper-client/issues/215>
6. *13 December 2014, Errors during Jasper installation*  
<https://github.com/jasperproject/jasper-client/issues/263>
7. *11 April 2015, Errors during Jasper installation*  
<https://github.com/jasperproject/jasper-client/issues/325>
8. *Errors during Spotify installation*  
<https://github.com/jasperproject/jasper-client/issues?utf8=%E2%9C%93&q=spotify>
9. *30 July 2014, Mopidy errors during installation*  
<https://github.com/mopidy/mopidy/issues/773>
10. *3 July 2014, Mopidy errors during installation*  
<https://github.com/kingosticks/mopidy-tunein/issues/8>
11. *31 July 2016, How to configure MPD server on Linux Software*  
<https://linuxconfig.org/configuring-the-mpd-music-server-on-ubuntu-linux>
12. *October 2014, Errors during Spotify installation*  
<https://discourse.mopidy.com/t/spotify-config-file-error/273/7>
13. *Mopidy Documentation*  
<https://docs.mopidy.com/en/latest/installation/raspberrypi/>
14. *Mopidy documentation on Debian*  
<https://docs.mopidy.com/en/latest/installation/debian/#debian-install>

15. *March 2015, Error during Spotify – mopidy installation*  
<https://discourse.mopidy.com/t/spotify-config-file-error/273/13>
16. *27 July 2016, Error during Spotify – mopidy installation*  
<https://github.com/mopidy/mopidy-spotify/issues/111>
17. *February 2015, Alsa mixer audio output on Raspberry Pi*  
<https://stackoverflow.com/questions/27053373/raspberry-alsa-sound-output-input-slave>
18. *2 April 2017, Error during Jasper installation*  
<https://github.com/jasperproject/jasper-client/issues/610>
19. *5 April 2017, Error during Jasper installation*  
<https://github.com/jasperproject/jasper-client/issues/612>
20. *31 July 2017, Error during Jasper installation*  
<https://github.com/jasperproject/jasper-client/issues/631>
21. *Mopidy – Spotify authentication*  
<https://mopidy.com/authenticate/#spotify>
22. *Mopidy – Spotify extencion*  
<https://github.com/mopidy/mopidy-spotify>
23. *Mopidy documentation for Debian*  
<https://docs.mopidy.com/en/latest/installation/debian/>
24. *21 August 2016, Install Mopidy on Raspbian*  
<https://www.inpimention.com/installing-mopidy-raspberry-pi-2/>
25. *29 July 2017, web client for mopidy*  
<https://github.com/dirkgroenen/mopidy-mopify>
26. *25 May 2018, Spotify music server with Mopidy on Raspberry Pi*  
<https://makingstuffwork.net/technology/create-spotify-enabled-music-server-with-mopidy/>
27. *October 2014, Error on Mopidy installation – LSB Script Missing*  
<https://raspberrypi.stackexchange.com/questions/23610/cannot-find-a-lsb-script-for-mopidy>
28. *October 2014, Error on Mopidy installation – LSB Script Missing*  
<https://duckduckgo.com/?q=update->

[rc.d%3A+error%3A+cannot+find+a+LSB+script+for+mopidy&t=raspberrypi&ia=web](https://www.raspberrypi.org/forums/viewtopic.php?p=409367)

29. 22 August 2013, Mopidy error no. 98 Adress already in use  
<https://www.raspberrypi.org/forums/viewtopic.php?p=409367>
30. Music Player Daemon on Mopidy  
<https://docs.mopidy.com/en/latest/ext/mpd/>
31. Run Mopidy as system service  
<https://docs.mopidy.com/en/latest/service/#service>
32. Mopidy configuration  
<https://docs.mopidy.com/en/latest/config/#config>
33. Mopidy discussion  
<https://discourse.mopidy.com/>
34. 2016, Mopidy and Spotify errors during installation  
[https://www.reddit.com/r/unixporn/comments/3ks4t6/ncmpcpp\\_mopidy\\_spotify\\_help/](https://www.reddit.com/r/unixporn/comments/3ks4t6/ncmpcpp_mopidy_spotify_help/)
35. 23 December 2012, MPD installation on the Raspberry Pi  
<http://www.raspyfi.com/ho-to-install-mpd-on-an-existing-debian-installation-on-raspberry-pi/>
36. October 2014, Spotify configuration errors  
<https://discourse.mopidy.com/t/spotify-config-file-error/273/3>
37. December 2014, Change Mopidy configuration  
<https://discourse.mopidy.com/t/impossible-to-change-mopidy-conf/506>
38. 3 January 2017, Mic setup on the Raspberry Pi  
<https://www.quora.com/How-do-I-set-up-a-microphone-on-Raspberry-pi-And-how-to-get-an-input-from-it>

## ΠΗΓΕΣ

1. A. Wager, Frances Wickham Lee and John P. Glaser. *Managing health care information systems: A practical approach for health care executives (I)*, Jossey - Bass: A Wiley imprint San Francisco, 2005
2. R. Agarwal, Y. Muthusamy and V. Viswanathan. *Voice Browsing the Web for Information Access*, Media Technologies Lab, Texas, XX  
<http://www.w3.org/Voice/1998/Workshop/RajeevAgarwal.html>
3. D. Sawant, P. Phadtare, C. Vyas. *Intelligent Voice Operating System (IVOS)*, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY, 2015
4. P. Pathak, *Speech Recognition Technology: Applications & Future*, International Journal of Advanced Computer Research, 2010
5. S. Russel, P. Norvig, *Artificial Intelligence: A Modern Approach*, 3<sup>rd</sup> Edition, Prentice Hall, 2009
6. D. Hand, H. Manilla, P. Smyth, *Principles of Data Mining*, The MIT, Press, 2001
7. J. Ticknor, *A Bayesian regularized artificial neural network for stock market forecasting*, Expert Systems with Applications, 2013
8. S. Singh, *Natural language processing for Information Extraction*, Department of Computing, Faculty of Science and Engineering, Macquarie University, 2018
9. S. Doan, M. Conway, T.M. Phuong, L. Ohno-Machado, *Natural language processing in biomedicine: A unified system architecture overview*, 2014
10. A. Pak, P. Paroubek, *Twitter as a corpus for sentiment analysis and opinion mining*, in, Proceedings of the Seventh International Conference on Language Resources and Evaluation, 2010
11. J. Wiebe, T. Wilson, C. Cardie, *Annotating expressions of opinions and emotions in language*, Kluwer Academic Publishers, 2005
12. M. Ogneva, *How Companies Can Use Sentiment Analysis to Improve Their Business*, 2010  
<https://mashable.com/2010/04/19/sentiment-analysis/>

13. S. Olatunji, M. Al-Ahmadi, M. Elshafei, Y. Fallatah, *Saudi Arabia Stock Prices Forecasting Using Artificial Neural Networks*, International Conference on Future Computer Sciences and Application, 2011
14. S. Simon and A. Raoot, *Accuracy Driven Artificial Neural Networks in Stock Market Prediction*, International Journal on Soft Computing (IJSC), 2012
15. D. Das, M. Shorif Uddin, *Data mining and Neural network Techniques in Stock Market Prediction: A Methodological Review*, International Journal of Artificial Intelligence & Applications, 2013
16. WWDC, Apple Developer 2015, videos  
<https://developer.apple.com/videos/wwdc2015/>
17. S. King, *Apple details how it rebuilt siri on mesos*, The Loop, 2015  
<https://www.loopinsight.com/2015/04/27/apple-details-how-it-rebuilt-siri-on-mesos/>
18. *What is Mesos? A distributed systems kernel*, Apache Mesos  
<http://mesos.apache.org/>
19. *Apple Rebuilds Siri Backend Services Using Apache Mesos*, InfoQ, 2019  
<https://www.infoq.com/news/2015/05/mesos-powers-apple-siri/>
20. *Siri for Developers*, Apple, 2019  
<https://developer.apple.com/siri/>
21. *Sirikit, Handle user requests for your app's services that originate from Siri or Maps*, Apple, 2019  
<https://developer.apple.com/documentation/sirikit>
22. *Cortana Skills Kit*, Microsoft, 2019  
<https://docs.microsoft.com/en-us/cortana/skills/overview>
23. N. Bostrom, *Superintelligence: Paths, dangers, strategies*, Oxford University Press, 2014
24. *Discovering and enacting the path to safe artificial general intelligence*, Open AI, 2019  
<https://openai.com/>