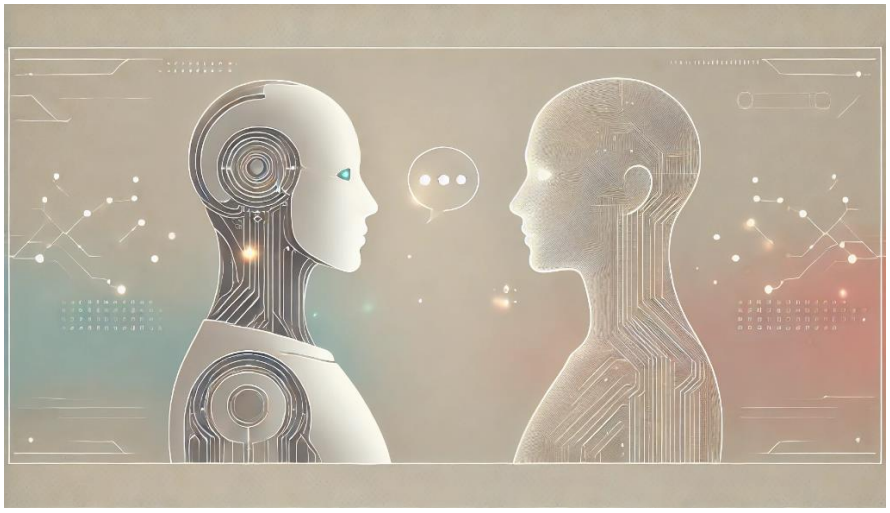
ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Ανάπτυξη διαλογικού συστήματος με ενσωμάτωση chatbot τεχνητής νοημοσύνης»



Του φοιτητή
Παλαιογιάννη Αθανασίου
Αρ. Μητρώου: 2019217

Επιβλέπων
Δρ. Τσιακμάκης Κυριάκος

Ημερομηνία 13/1/2025

Τίτλος Δ.Ε.: Ανάπτυξη διαλογικού συστήματος με ενσωμάτωση chatbot τεχνητής νοημοσύνης

Κωδικός Δ.Ε.: 24186

Ονοματεπώνυμο φοιτητή: Παλαιογιάννης Αθανάσιος

Ονοματεπώνυμο εισηγητή: Τσιακμάκης Κυριάκος

Ημερομηνία ανάληψης Δ.Ε.: 9/5/2024

Ημερομηνία περάτωσης Δ.Ε.: 13/1/2025

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραιτήτως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Σε αυτούς που μου συμπαραστάθηκαν στα φοιτητικά μου χρόνια»

Πρόλογος

Το ενδιαφέρον μου για την τεχνητή νοημοσύνη και τις πιθανές εφαρμογές της στην αλληλεπίδραση ανθρώπου-υπολογιστή ήταν ο λόγος που επιλέχθηκε αυτή η διατριβή. Η εργασία αυτή μου έδωσε την ευκαιρία να εξετάσω πιο προσεκτικά τις τεχνολογίες Speech-to-Text και Text-to-Speech και να τις ενσωματώσω σε ένα διαδραστικό πλαίσιο με ανθρωποειδή ρομπότ. Κατά τη διάρκεια της διαδικασίας σχεδιασμού και υλοποίησης, ανέπτυξα πρακτικές δεξιότητες στον προγραμματισμό και τη χρήση σύγχρονων APIs όπως το Deepgram κα το ElevenLabs, μαζί με τη διαχείριση φυσικής γλώσσας μέσω του ChatGPT.

Η διπλωματική εργασία πρότεινε μια νέα πρόκληση που επέτρεψε την ανάπτυξη των γνώσεών μου σχετικά με τον τρόπο εφαρμογής της τεχνητής νοημοσύνης σε πραγματικά σενάρια. Η πολυγλωσσική ικανότητα και η χαμηλή καθυστέρηση απόκρισης αύξησαν σημαντικά τη λειτουργικότητα αυτού του έργου, καθιστώντας το πολύ χρήσιμο σε ένα εκπαιδευτικό, επαγγελματικό και κοινωνικό περιβάλλον. Πέρα από τη μάθηση που συνάντησα, αυτό με βοήθησε να ενισχύσω την κριτική μου σκέψη και τις δεξιότητες διαχείρισης έργων.

Περίληψη

Η παρούσα διατριβή αφορά την ανάπτυξη ενός διαδραστικού συστήματος που ενσωματώνει τεχνητή νοημοσύνη χρησιμοποιώντας Speech-to-Text, Text-to-Speech και Natural Language Processing ως κύρια τεχνολογικά δομικά στοιχεία. Το τελικό σύστημα θα εφαρμοστεί σε ανθρωποειδές ρομπότ προκειμένου να προσδώσει μεγαλύτερη ρευστότητα στην αλληλεπίδραση μεταξύ ανθρώπων και μηχανών. Με τη χρήση αυτών των τεχνολογιών, επιτυγχάνει τα καθήκοντα της αναγνώρισης και της σύνθεσης ομιλίας με λεπτομέρεια, μαζί με την κατανόηση φυσικής γλώσσας που είναι ακριβής και γρήγορη.

Κατά την υλοποίηση του έργου χρησιμοποιήθηκαν διεπαφές προγραμματισμού εφαρμογών APIs όπως το Deepgram για Speech-to-Text και το ElevenLabs για Text-to-Speech, ενώ η επεξεργασία φυσικής γλώσσας έγινε μέσω του ChatGPT. Η αρχιτεκτονική περιλαμβάνει δυνατότητες ροής, οι οποίες έχουν σχεδιαστεί για να μειώνουν την καθυστέρηση προσομοιώνοντας έτσι έναν πραγματικό διάλογο. Ο σχεδιασμός και η ανάπτυξη βασίστηκαν σε μια εγκατάσταση Raspberry Pi που περιλαμβάνει ένα μικρόφωνο, ένα ηχείο και μια εξωτερική κάρτα ήχου για τη μεγιστοποίηση της απόδοσης.

Τα αποτελέσματα του έργου καταδεικνύουν ότι το σύστημα είναι αποτελεσματικό στο χειρισμό συνομιλιών σε πραγματικό χρόνο, με χαμηλή καθυστέρηση στην ανταπόκριση και ακριβή αναγνώριση και παραγωγή ομιλίας. Το έργο αυτό δείχνει τις δυνατότητες χρήσης αυτού του συστήματος σε σχολεία, σε ρομπότ που συνεργάζονται μεταξύ τους, ακόμη και στην εξυπηρέτηση πελατών, ώστε να γίνουν τα ρομποτικά συστήματα πιο εύχρηστα και ευφυή. Εν κατακλείδι, το έργο αυτό συμβάλλει στην ανάπτυξη της έρευνας στην τεχνητή νοημοσύνη και την επεξεργασία φυσικής γλώσσας καθώς ανοίγει νέους τρόπους χρήσης αυτών των τεχνολογιών στην καθημερινή ζωή.

«Development of a dialog system with integration of artificial intelligence chatbot»

Thanasis Palaiogiannis

Abstract

This thesis concerns the development of an interactive system that incorporates artificial intelligence using Speech-to-Text, Text-to-Speech and Natural Language Processing as main technological building blocks. The final system will be applied to a humanoid robot to give more fluidity to the interaction between humans and machines. Using these technologies, it effectively performs speech recognition and synthesis in detail, along with natural language understanding that is accurate and fast.

Application programming interfaces APIs such as Deepgram for Speech-to-Text and ElevenLabs for Text-to-Speech were used in the implementation of the project, while natural language processing was done through ChatGPT. The architecture includes streaming capabilities designed to reduce latency by supporting multiple languages simultaneously. The design and development were based on a Raspberry Pi setup that included a microphone, a speaker and an external sound card to maximize performance.

The results of the project demonstrate that the system is effective in handling real-time conversations, with low response latency and accurate speech recognition and speech generation. This project shows the potential of using this system in schools, in robots working together, and even in customer service to make robotic systems more usable and intelligent. In conclusion, this project contributes to the development of research in artificial intelligence and natural language processing as it opens up new ways of using these technologies in everyday life.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον καθηγητή μου κύριο Τσιακμάκη Κυριάκο για την έμπρακτη προσφορά και καθοδήγηση του, τους γονείς μου για όλη την υποστήριξη και την κοπέλα μου που βοήθησε στην ορθότητα της συγγραφής αυτής διπλωματικής.

Περιεχόμενα

Κατάλογος Σχημάτων

Συντομογραφίες

STT	Speech To Text
NLP	Natural Language Processing
TTS	Text To Speech
GPT	Generative Pre-trained Transformer
AI	Artificial Intelligence
API	Application Programming Interface
HI	Human Intelligence
NLU	Natural Language Understanding
Δ.Ε	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
HMI	Human Machine Interaction

Κεφάλαιο 1ο: Εισαγωγή

1.1 Εισαγωγή

Η ανάπτυξη διαδραστικών συστημάτων που ενσωματώνουν τεχνολογίες Τεχνητής Νοημοσύνης (Artificial Intelligence, AI) είναι ένας εξαιρετικά καινοτόμος και ταχέως εξελισσόμενος τομέας της επιστήμης των υπολογιστών. Η αλληλεπίδραση ανθρώπου-μηχανής (Human-Machine Interaction, HMI) έχει αναδειχθεί σε κεντρική απαίτηση για την ανάπτυξη συστημάτων που κατανοούν και ανταποκρίνονται στη φυσική γλώσσα. Οι εφαρμογές των τεχνολογιών αυτών βρίσκονται σε όλο και περισσότερους τομείς, όπως η εξυπηρέτηση πελατών, η ρομποτική, η υγειονομική περίθαλψη, η εκπαίδευση και η ψυχαγωγία.

Η παρούσα διατριβή ασχολείται με τη δημιουργία ενός διαδραστικού συστήματος που μπορεί να εφαρμοστεί σε τεχνολογίες Speech-to-Text (STT) και Text-to-Speech (TTS), σε συνδυασμό με πλατφόρμες τεχνητής νοημοσύνης όπως η ChatGPT. Το σύστημα προορίζεται για χρήση σε περιβάλλοντα ανθρωποειδών ρομπότ, προκειμένου να παρέχει ένα ευέλικτο πλαίσιο για φυσικές αλληλεπιδράσεις ανθρώπου-μηχανής. Το σύστημα αυτό ενσωματώνει όλες αυτές τις τεχνολογίες για την επίτευξη ενός νέου περιβάλλοντος διαλόγου, αυξάνοντας τόσο τη χρηστικότητα όσο και την ευφυΐα των ρομποτικών συστημάτων.

Εφαρμόζει προηγμένους αλγορίθμους NLP, επιτρέποντας στο σύστημα να κατανοεί και να παράγει με ακρίβεια φυσική γλώσσα. Υιοθετεί, παράλληλα, προσεγγίσεις για τη βελτίωση της ευχέρειας και της αποδοτικότητας του συστήματος, ενσωματώνοντας την τεχνολογία STT ροής, προκειμένου να μειωθεί η καθυστέρηση. Το σύστημα ικανοποιεί τις απαιτήσεις της διαδραστικής επικοινωνίας σε πραγματικό χρόνο και ταυτόχρονα ενσωματώνει στρατηγικές που θα συνεισφέρουν στην προσομοίωση πραγματικού διαλόγου.

Η επιλογή αυτών των τεχνολογιών βασίζεται στις τρέχουσες εξελίξεις στον τομέα της τεχνητής νοημοσύνης και του NLP. Συγκεκριμένα, οι τεχνολογίες STT και TTS έχουν καταστήσει δυνατή την αμφίδρομη επικοινωνία ανθρώπου-μηχανής, όπου οι μηχανές μπορούν να αποκτήσουν τη δυνατότητα να ακούν και να απαντούν με τρόπο φυσικό για τον άνθρωπο. Ομοίως, οι δυνατότητες NLP επιτρέπουν την κατανόηση και την ανάλυση του περιεχομένου φυσικής γλώσσας, επιτρέποντας στα συστήματα να αναγνωρίζουν τις ανάγκες των χρηστών και να παρέχουν προσαρμοσμένες απαντήσεις.

1.1.1 Πρόκληση και Σημασία της Εργασίας

Παρά την πρόοδο που έχει σημειωθεί, η εφαρμογή αυτών των τεχνολογιών σε συστήματα πραγματικού χρόνου -όπως τα ανθρωποειδή ρομπότ- παραμένει μια πρόκληση. Τέτοια συστήματα απαιτούν χαμηλό χρόνο απόκρισης, υψηλή ακρίβεια των απαντήσεων και προσαρμοστικότητα σε πολύγλωσσα

περιβάλλοντα. Η παρούσα εργασία προσπαθεί να καλύψει αυτό το κενό με την υλοποίηση ενός συστήματος που ενσωματώνει τεχνολογίες streaming STT για τη μείωση του χρόνου επεξεργασίας, και δυνατότητες NLP που υποστηρίζουν διαδραστική επικοινωνία στα αγγλικά καθώς οι αλγόριθμοι είναι πιο εξοικειωμένοι στην αναγνώριση αυτής της γλώσσας.

Το προτεινόμενο σύστημα συνδυάζει τις εξής καινοτομίες:

- Ενσωμάτωση τεχνολογιών streaming STT και TTS: Χρήση σύγχρονων APIs, όπως το Deepgram για την άμεση αναγνώριση φωνής και το ElevenLabs για την παραγωγή φυσικής φωνής.
- Διαχείριση φυσικής γλώσσας μέσω ChatGPT: Ενσωμάτωση ενός ισχυρού αλγορίθμου NLP, που προσφέρει τη δυνατότητα επεξεργασίας ερωτήσεων και δημιουργίας φυσικών, λεπτομερών απαντήσεων.
- Εφαρμογή σε ανθρωποειδή ρομπότ: Σχεδίαση και υλοποίηση ενός συστήματος που προσαρμόζεται στις απαιτήσεις φυσικής παρουσίας και διαδραστικότητας ρομποτικών συστημάτων.

1.1.2 Εφαρμογές και Επιπτώσεις

Η σημασία της παρούσας εργασίας εντοπίζεται στην προώθηση της ευφυούς αλληλεπίδρασης ανθρώπου-μηχανής. Οι πιθανές εφαρμογές του προτεινόμενου συστήματος περιλαμβάνουν:

- Εκπαίδευση: Ρομπότ που λειτουργούν ως διαδραστικοί εκπαιδευτές, προσαρμόζοντας το περιεχόμενο των μαθημάτων στις ανάγκες των μαθητών.
- Εξυπηρέτηση πελατών: Διαλογικά ρομπότ που παρέχουν υποστήριξη σε πραγματικό χρόνο σε επιχειρήσεις και οργανισμούς.
- Υγεία: Βοηθοί που υποστηρίζουν ασθενείς σε διαδικασίες αυτοεξυπηρέτησης ή παρέχουν ψυχολογική υποστήριξη.
- Ρομποτική συνεργασία: Ανθρωποειδή ρομπότ που συνεργάζονται με ανθρώπους σε βιομηχανικές ή ερευνητικές διαδικασίες.

Συνοψίζοντας, η εργασία αυτή στοχεύει στην συμβολή εμβάθυνσης της έρευνας γύρω από τη φυσική γλώσσα, την τεχνητή νοημοσύνη και τις εφαρμογές τους στην καθημερινή ζωή, επιλύοντας πρακτικές προκλήσεις και ανοίγοντας τον δρόμο για νέες δυνατότητες.

1.2 Στόχος της διατριβής

Οι κύριοι στόχοι της παρούσας εργασίας συνοψίζονται ως εξής:

- Ανάπτυξη ενός λειτουργικού συστήματος διαλόγου: Το σύστημα καθίσταται αναγκαίο να έχει την δυνατότητα να κατανοεί και να απαντά σε φυσική γλώσσα, κυρίως στα αγγλικά.
- Ενσωμάτωση τεχνολογιών STT και TTS: Χρήση προηγμένων εργαλείων και APIs, όπως το Deepgram και το ElevenLabs, για την υλοποίηση αποτελεσματικής μετατροπής ήχου σε κείμενο και κειμένου σε ήχο.
- Διασφάλιση χαμηλού χρόνου απόκρισης: Χρήση τεχνικών streaming για την ελαχιστοποίηση καθυστερήσεων.
- Δημιουργία ενός ευφυούς διαλογικού περιβάλλοντος: Ενσωμάτωση του ChatGPT για την παραγωγή φυσικών και εύστοχων απαντήσεων.
- Προσαρμογή σε ρομποτικά περιβάλλοντα: Σχεδιασμός συστήματος που να μπορεί να εφαρμοστεί σε ανθρωποειδή ρομπότ.

1.3 Δομή της εργασίας

Η παρούσα εργασία είναι δομημένη ως εξής:

- Κεφάλαιο 1: Παρουσιάζονται η εισαγωγή, οι στόχοι της έρευνας, καθώς και η δομή που ακολουθείται. Περιγράφονται επίσης τα γενικά πλαίσια και η σημασία της μελέτης.
- Κεφάλαιο 2: Περιλαμβάνει τη βιβλιογραφική ανασκόπηση με έμφαση στις βασικές έννοιες και τεχνολογίες που σχετίζονται με το αντικείμενο της εργασίας, όπως οι τεχνολογίες Speech-to-Text (STT), Text-to-Speech (TTS) και Natural Language Processing (NLP). Επιπλέον, γίνεται αναφορά σε υπάρχουσες εφαρμογές και λύσεις.
- Κεφάλαιο 3: Περιγράφεται αναλυτικά ο σχεδιασμός και η υλοποίηση του συστήματος. Περιλαμβάνει την αρχιτεκτονική του συστήματος, τα εργαλεία και τις τεχνολογίες που χρησιμοποιήθηκαν, καθώς και τις διαδικασίες ανάπτυξης.
- Κεφάλαιο 4: Παρουσιάζονται τα πειραματικά αποτελέσματα και η αξιολόγηση του συστήματος. Γίνεται συζήτηση σχετικά με την αποτελεσματικότητα και τις επιδόσεις της υλοποίησης, συνοδευόμενη από ανάλυση δεδομένων και γραφήματα.
- Κεφάλαιο 5: Συμπεράσματα της έρευνας, οι περιορισμοί της εργασίας και προτάσεις για μελλοντική έρευνα ή ανάπτυξη.

Κεφάλαιο 2ο: Θεωρητικό Υπόβαθρο

2.1 Εισαγωγή

Η ανάπτυξη ενός διαλογικού συστήματος που ενσωματώνει τεχνολογίες όπως Speech-to-Text (STT), Natural Language Processing (NLP) και Text-to-Speech (TTS) απαιτεί βαθιά κατανόηση των βασικών θεωρητικών πλαισίων και εργαλείων που υποστηρίζουν τη λειτουργία του. Το παρόν κεφάλαιο παρέχει το θεωρητικό υπόβαθρο που είναι απαραίτητο για την κατανόηση της διπλωματικής εργασίας, εστιάζοντας στις τεχνολογίες που υλοποιούνται, τις αρχές λειτουργίας τους και τα πλεονεκτήματα που προσφέρουν στη δημιουργία ενός σύγχρονου διαλογικού βοηθού.

Οι τεχνολογίες επεξεργασίας φυσικής γλώσσας (NLP) έχουν σημειώσει εντυπωσιακή πρόοδο τα τελευταία χρόνια, επιτρέποντας στα υπολογιστικά συστήματα να επεξεργάζονται και να κατανοούν τη γλώσσα με τρόπους που προσεγγίζουν την ανθρώπινη αντίληψη. Σε συνδυασμό με την αναγνώριση φωνής (STT), η οποία μετατρέπει τη φωνή σε κείμενο, και την παραγωγή φωνής (TTS), η οποία μετατρέπει το κείμενο σε φυσικό ήχο, καθίσταται δυνατή η δημιουργία ολοκληρωμένων συστημάτων αλληλεπίδρασης ανθρώπου-μηχανής.

Το κεφάλαιο αυτό είναι δομημένο με στόχο να παρουσιάσει τις θεωρητικές αρχές που διέπουν τις τεχνολογίες αυτές, ενώ παράλληλα εστιάζει στις εφαρμογές που ενσωματώθηκαν στο συγκεκριμένο έργο. Αρχικά, παρουσιάζεται η έννοια και η εξέλιξη της επεξεργασίας φυσικής γλώσσας, καθώς και οι θεμελιώδεις έννοιες της τεχνητής νοημοσύνης που σχετίζονται με το NLP. Στη συνέχεια, εξετάζονται αναλυτικά οι τεχνολογίες Speech-to-Text και Text-to-Speech, με ιδιαίτερη έμφαση στις δυνατότητες και τα πλεονεκτήματα των APIs που χρησιμοποιήθηκαν, όπως το Deepgram, το OpenAI και το ElevenLabs.

Παράλληλα, αναλύεται ο ρόλος των APIs και των βιβλιοθηκών προγραμματισμού Python στη διασύνδεση των επιμέρους τεχνολογιών, προσφέροντας μια σφαιρική εικόνα του θεωρητικού πλαισίου. Η παρουσίαση των εννοιών αυτών δεν περιορίζεται μόνο στη θεωρία, αλλά συνδέεται άμεσα με τις πρακτικές προεκτάσεις της εφαρμογής τους στο έργο.

Η κατανόηση του θεωρητικού πλαισίου που περιγράφεται στο κεφάλαιο αυτό είναι καθοριστική για τη συνολική αξιολόγηση του συστήματος που αναπτύχθηκε, καθώς και για την εκτίμηση της συμβολής του στην ευρύτερη ερευνητική κοινότητα.

2.2 Ανθρώπινη νοημοσύνη

Η ανθρώπινη νοημοσύνη (Human Intelligence-HI) αποτελεί μία από τις πιο πολύπλοκες και δυναμικές έννοιες, η οποία συνδέεται άμεσα με την ικανότητα κατανόησης, προσαρμογής και επίλυσης προβλημάτων. Η έννοια αυτή δεν περιορίζεται μόνο στην απόκτηση γνώσεων, αλλά περιλαμβάνει και

την ικανότητα χρήσης τους σε ποικίλες συνθήκες. Πρόκειται για ένα πολυδιάστατο φαινόμενο που συνδυάζει γνωστικές, συναισθηματικές και κοινωνικές δεξιότητες.

Η ανθρώπινη μάθηση, ένα από τα κύρια χαρακτηριστικά της νοημοσύνης, βασίζεται σε διαδραστικές διαδικασίες που επηρεάζονται από το περιβάλλον. Η εκμάθηση, τόσο σε επίπεδο γεγονότων όσο και αξιών, χαρακτηρίζεται από δυναμική προσαρμογή, όπου ο ανθρώπινος νους αποθηκεύει και αναδιαμορφώνει πληροφορίες ανάλογα με τις εμπειρίες του. Αυτή η διαδικασία οδηγεί στην αυτοπροσαρμογή της μνήμης, η οποία μεταβάλλεται συνεχώς ανάλογα με το ανθρώπινο-υπολογιστικό περιβάλλον και επιτρέπει την αναγνώριση χαρακτηριστικών που προηγουμένως δεν είχαν παρατηρηθεί.

Μια ουσιώδης διάσταση της ανθρώπινης νοημοσύνης είναι η ικανότητα δημιουργίας εννοιών και κατανόησής τους. Οι άνθρωποι μπορούν να επεξεργάζονται αφηρημένες έννοιες, να σχηματίζουν συνδέσεις και να παράγουν δημιουργικές λύσεις, χαρακτηριστικά που διαφέρουν ριζικά από τις μηχανές. Ενώ οι μηχανές βασίζονται σε προγραμματισμένες λογικές διαδικασίες, οι άνθρωποι έχουν τη δυνατότητα να αξιοποιούν την ηθική, τη συνείδηση και την ευαισθησία στις αποφάσεις τους.

Επιπλέον, η φυσική νοημοσύνη ενσωματώνει δύο βασικές έννοιες: την τυπική έννοια, η οποία επικεντρώνεται στη λογική δράση και στη μεγιστοποίηση της χρησιμότητας σε συνθήκες περιορισμένων πόρων, και την ουσιαστική έννοια, η οποία εστιάζει σε αξίες και συναισθηματική ευαισθησία. Αυτή η διάσταση αντικατοπτρίζει την πολυπλοκότητα της ανθρώπινης συμπεριφοράς, η οποία επηρεάζεται όχι μόνο από τον ορθολογισμό αλλά και από παράγοντες όπως η ηθική και η κοινωνική αντίληψη.

Η κατανόηση της ανθρώπινης νοημοσύνης αποτελεί τη βάση για την ανάπτυξη της τεχνητής νοημοσύνης, καθώς εμπνέει τη δημιουργία συστημάτων που προσομοιώνουν τις ανθρώπινες ικανότητες. Η συνειδητοποίηση των περιορισμών και των μοναδικών χαρακτηριστικών της φυσικής νοημοσύνης είναι κρίσιμη για την εξέλιξη των τεχνολογιών που στοχεύουν στην ενίσχυση της ανθρώπινης ζωής. [1]

2.3 Τεχνητή Νοημοσύνη

Η έννοια της Τεχνητής Νοημοσύνης (Artificial Intelligence - AI) μπορεί να αναλυθεί σε δύο βασικά σκέλη: το "τεχνητό" και τη "νοημοσύνη". Το "τεχνητό" αφορά ένα ανθρώπινα κατασκευασμένο σύστημα, ενώ ο όρος "νοημοσύνη" είναι πιο πολύπλοκος και συχνά αναφέρεται στην έρευνα ανθρώπινων γνωστικών δραστηριοτήτων. Αυτές περιλαμβάνουν έννοιες όπως η συνείδηση, η σκέψη και ο συλλογισμός. Η Τεχνητή Νοημοσύνη, ως επιστημονικό πεδίο, συνδυάζει την ανάπτυξη και προσομοίωση της ανθρώπινης νοημοσύνης μέσω θεωριών, μεθόδων και τεχνολογιών, με σκοπό την εφαρμογή αυτών σε πρακτικά συστήματα.

Η AI αποτελεί κλάδο της επιστήμης των υπολογιστών, γνωστή επίσης ως μηχανική νοημοσύνη, και αποσκοπεί στο να δώσει στους υπολογιστές τη δυνατότητα να προσομοιώνουν ανθρώπινες γνωστικές

λειτουργίες. Αυτές περιλαμβάνουν τη μάθηση, τον συλλογισμό, τη σκέψη και τον σχεδιασμό. Ο στόχος είναι οι υπολογιστές να αποκτήσουν χαρακτηριστικά αντίστοιχα της ανθρώπινης νοημοσύνης, ενισχύοντας την ικανότητά τους να επιλύουν πολύπλοκα προβλήματα σε διαφορετικούς τομείς εφαρμογής. Αυτό περιλαμβάνει τη βελτίωση της ανθρώπινης ζωής, τόσο σε επαγγελματικό όσο και σε προσωπικό επίπεδο, μέσω της αντικατάστασης ανθρώπων σε επικίνδυνες, δύσκολες ή πολύπλοκες εργασίες.

Η Τεχνητή Νοημοσύνη αποτελεί ένα διεπιστημονικό πεδίο, το οποίο αντλεί στοιχεία από την επιστήμη των υπολογιστών, τη φιλοσοφία, τη γλωσσολογία, την ψυχολογία και πολλές άλλες επιστήμες. Πρόκειται για μια ενσωματωμένη προσέγγιση που καλύπτει ένα ευρύ φάσμα φυσικών και κοινωνικών επιστημών. Ειδικότερα, η AI συνδέεται στενά με την ανάπτυξη της επιστήμης της σκέψης, καθώς αποτελεί την πρακτική εφαρμογή αυτής. Η σχέση της με την επιστήμη των υπολογιστών είναι επίσης θεμελιώδης, καθώς η ανάπτυξη των δικτύων υπολογιστών επηρεάζει άμεσα την εξέλιξή της.

Τέλος, η Τεχνητή Νοημοσύνη δεν περιορίζεται πλέον στην απλή επεξεργασία δεδομένων, αλλά προχωρά στην επεξεργασία γνώσης, οδηγώντας σε σημαντικές καινοτομίες. Η εξέλιξή της επηρεάζει όχι μόνο την τεχνολογία, αλλά και την ίδια την κατανόηση του ανθρώπινου γνωστικού δυναμικού. [2]

2.4 Τεχνητή νοημοσύνη και ανθρώπινος συλλογισμός

Η σύγκριση και η σύνδεση μεταξύ της ανθρώπινης νοημοσύνης (Human Intelligence - HI) και της τεχνητής νοημοσύνης (Artificial Intelligence - AI) αναδεικνύουν τα διαφορετικά χαρακτηριστικά και τις δυνατότητές τους, καθώς και τις προκλήσεις που αντιμετωπίζουν. Η ανθρώπινη νοημοσύνη ξεχωρίζει για τη δημιουργικότητα, την ικανότητα κατανόησης συναισθημάτων και τη διαχείριση αβέβαιων καταστάσεων, ενώ η τεχνητή νοημοσύνη διακρίνεται για την ταχύτητα και την αποδοτικότητά της σε συγκεκριμένα καθήκοντα που απαιτούν μεγάλη υπολογιστική ισχύ .

Η εκμάθηση γνώσης αποτελεί θεμελιώδη πτυχή της ανθρώπινης νοημοσύνης. Το HI μπορεί να λειτουργεί σε περιβάλλοντα με υψηλό θόρυβο και να χρησιμοποιεί διαισθητική σκέψη για τη λήψη αποφάσεων, ενώ έχει τη δυνατότητα να μαθαίνει και να προσαρμόζεται ακόμα και σε καταστάσεις με ελλιπή δεδομένα εκπαίδευσης. Αντίθετα, το AI βασίζεται σε καλά ορισμένα και πλήρως επισημασμένα σύνολα δεδομένων για την εκπαίδευσή του. Η ποιότητα της τεχνητής νοημοσύνης εξαρτάται από την ποιότητα αυτών των δεδομένων, γεγονός που περιορίζει την ικανότητά της να διαχειρίζεται μη επαρκώς καθορισμένες καταστάσεις. Ωστόσο, ενώ η ανθρώπινη νοημοσύνη υπερέχει στη δημιουργία καινοτόμων ιδεών και στη διαχείριση αβεβαιότητας, το AI αποδίδει εξαιρετικά σε περιβάλλοντα που απαιτούν υψηλή ταχύτητα, επαναληπτικότητα και ακρίβεια .

Η λήψη αποφάσεων είναι μια άλλη πτυχή όπου τόσο το HI όσο και το AI αντιμετωπίζουν προκλήσεις. Η ανθρώπινη νοημοσύνη μπορεί να επηρεάζεται από προκαταλήψεις και στερεότυπα, ενώ η τεχνητή νοημοσύνη μπορεί να επηρεάζεται από τα δεδομένα εκπαίδευσης, τα οποία ενδέχεται να είναι

μεροληπτικά. Παρόλα αυτά, η ανθρώπινη νοημοσύνη μπορεί να προσαρμόζεται καλύτερα σε νέες πληροφορίες και να λαμβάνει αποφάσεις με βάση μη μετρήσιμους παράγοντες, όπως η ηθική και η ευαισθησία .

Η εκτέλεση εργασιών παρουσιάζει επίσης διαφορές. Το AI έχει τη δυνατότητα να απομνημονεύει και να επεξεργάζεται τεράστιο όγκο δεδομένων, προσφέροντας συνέπεια και αποδοτικότητα. Από την άλλη πλευρά, η ανθρώπινη νοημοσύνη ξεχωρίζει για την ικανότητά της να κατανοεί συναισθήματα, σχέσεις και μοτίβα, καθώς και για τη διαχείριση σύνθετων και αβέβαιων καταστάσεων. Αυτές οι δεξιότητες καθιστούν το HI μοναδικό για εφαρμογές που απαιτούν δημιουργικότητα και ενσυναίσθηση. [3]

Η σχέση μεταξύ της ανθρώπινης και της τεχνητής νοημοσύνης βασίζεται στη στενή σύνδεση της δεύτερης με την ανθρώπινη λογική. Η έρευνα στον τομέα της τεχνητής νοημοσύνης εμπνέεται από τα δίκτυα των νευρώνων στον ανθρώπινο εγκέφαλο. Το AI χρησιμοποιεί επεξεργασία φυσικής γλώσσας και μηχανική μάθηση για να αναλύει τεράστιο όγκο δεδομένων σε πολλές γλώσσες, κάτι που η ανθρώπινη νοημοσύνη δεν μπορεί να επιτύχει με την ίδια ταχύτητα. Ωστόσο, η ανθρώπινη λογική παραμένει ένα σημαντικό πλεονέκτημα για την κατανόηση της ευπάθειας και της αλλαγής σε ένα διαρκώς εξελισσόμενο περιβάλλον. Στο μέλλον, η συνεργασία μεταξύ HI και AI αναμένεται να προσφέρει σημαντικές δυνατότητες, ενισχύοντας την ικανότητα αντιμετώπισης νέων προκλήσεων και την προσαρμογή σε καινοτόμες εφαρμογές .

Συνολικά, η κατανόηση των δυνατοτήτων και των αδυναμιών τόσο της ανθρώπινης όσο και της τεχνητής νοημοσύνης είναι κρίσιμη για την αξιοποίησή τους σε συνδυασμό. Ενώ η ανθρώπινη νοημοσύνη διαθέτει μοναδικές δεξιότητες, όπως η δημιουργικότητα και η διαίσθηση, η τεχνητή νοημοσύνη προσφέρει ταχύτητα, συνέπεια και επεξεργαστική ισχύ. Η συνεργασία μεταξύ των δύο τύπων νοημοσύνης μπορεί να οδηγήσει σε καινοτόμες λύσεις και σημαντική πρόοδο σε διάφορους τομείς. [4]

2.5 Επεξεργασία φυσικής γλώσσας (NLP)

Η Επεξεργασία Φυσικής Γλώσσας (NLP) αποτελεί έναν από τους πιο κρίσιμους κλάδους της τεχνητής νοημοσύνης, ο οποίος εστιάζει στη γεφύρωση της επικοινωνίας μεταξύ ανθρώπων και υπολογιστών μέσω της φυσικής γλώσσας. Ο κλάδος αυτός συνδυάζει την επιστήμη των υπολογιστών, τη γλωσσολογία, τη στατιστική και τη μηχανική μάθηση για να επιτύχει την κατανόηση, την ανάλυση και την παραγωγή κειμένων ή φωνής από υπολογιστικά συστήματα. Το NLP έχει εξελιχθεί σε ένα από τα θεμέλια της ανθρώπινης-υπολογιστικής αλληλεπίδρασης, καθιστώντας εφικτές τις εφαρμογές όπως η μηχανική μετάφραση, οι ψηφιακοί βοηθοί και τα συστήματα ανάλυσης συναισθημάτων [5] [6].

2.5.1 Ιστορική Εξέλιξη του NLP

Η ιστορία του NLP μπορεί να αναχθεί στις αρχές του 20ού αιώνα, όταν οι πρώτες προσπάθειες εστίασαν στη μηχανική μετάφραση και τη γραμματική ανάλυση. Κατά τη δεκαετία του 1980, η εισαγωγή στατιστικών μεθόδων και αλγορίθμων, όπως τα Hidden Markov Models, προσέφερε πιο αξιόπιστες λύσεις στην ανάλυση μεγάλων δεδομένων γλώσσας. Η επανάσταση στη βαθιά μάθηση, ειδικότερα με την εμφάνιση των νευρωνικών δικτύων και των μοντέλων Transformer, έθεσε νέες βάσεις για την ακρίβεια και την αποτελεσματικότητα των συστημάτων NLP. [5]

Κατά την τελευταία δεκαετία, η πρόοδος στα γλωσσικά μοντέλα μεγάλης κλίμακας, όπως τα GPT, συνέβαλε σημαντικά στην ικανότητα του NLP να αναπαράγει ανθρώπινη γλώσσα με εξαιρετική ακρίβεια. Τα μοντέλα αυτά χρησιμοποιούν εκατομμύρια παραμέτρους και εκπαιδεύονται σε τεράστια σύνολα δεδομένων, επιτρέποντας την κατανόηση της σημασιολογίας, της συντακτικής δομής και των συμφραζόμενων σε υψηλό επίπεδο. [6]

2.5.2 Διαδικασίες και Τεχνολογίες NLP

Το NLP περιλαμβάνει μια σειρά από διαδικασίες και τεχνολογίες που συνεργάζονται για την κατανόηση και την παραγωγή γλώσσας:

1. **Προεπεξεργασία Κειμένου**: Περιλαμβάνει την κανονικοποίηση δεδομένων, όπως η αφαίρεση σημείων στίξης, η μετατροπή λέξεων σε ρίζες τους (stemming) και η τυποποίηση του κειμένου για την εξάλειψη ασαφειών. Αυτή η διαδικασία επιτρέπει τη δημιουργία καθαρών δεδομένων για ανάλυση. [6]

2. **Ανάλυση Συντακτικής και Σημασιολογικής Δομής**: Η συντακτική ανάλυση εξετάζει τη γραμματική δομή προτάσεων, ενώ η σημασιολογική ανάλυση επικεντρώνεται στην κατανόηση του νοήματος και των συμφραζόμενων. Η τεχνολογία αυτή έχει εφαρμογές σε συστήματα μετάφρασης και σε συστήματα υποστήριξης λήψης αποφάσεων [5].

3. **Αναγνώριση Οντοτήτων (Named Entity Recognition - NER)**: Το NER βοηθά στην αναγνώριση και κατηγοριοποίηση σημαντικών οντοτήτων, όπως ονόματα, τοποθεσίες και ημερομηνίες. Αυτή η τεχνολογία είναι σημαντική για εφαρμογές εξαγωγής πληροφορίας από δεδομένα. [5] [6]

4. **Ανάλυση Συναισθήματος**: Επιτρέπει την ανίχνευση συναισθημάτων (θετικά, αρνητικά ή ουδέτερα) σε κείμενα, χρησιμοποιώντας τεχνικές βαθιάς μάθησης. Η τεχνολογία αυτή χρησιμοποιείται σε εφαρμογές όπως η ανάλυση δεδομένων πελατών. [6]

5. **Γλωσσικά Μοντέλα Μεγάλης Κλίμακας**: Τα μοντέλα όπως το GPT (Generative Pre-trained Transformer) ενσωματώνουν μηχανισμούς προσοχής για την κατανόηση μεγάλων ακολουθιών

κειμένου και τη δημιουργία ανθρώπινων απαντήσεων. Αυτά τα μοντέλα εκπαιδεύονται σε δισεκατομμύρια δεδομένων, επιτρέποντας την αυτοματοποίηση πολύπλοκων διαδικασιών. [2]

2.5.3 Εφαρμογές NLP

Το NLP εφαρμόζεται σε ένα ευρύ φάσμα τομέων, από την αυτόματη μετάφραση και τους ψηφιακούς βοηθούς έως τη δημιουργία περιεχομένου και τη συστηματική ανάλυση μεγάλων ποσοτήτων δεδομένων. Για παράδειγμα, τα σύγχρονα συστήματα υποστήριξης πελατών βασίζονται σε γλωσσικά μοντέλα για την κατανόηση αιτημάτων και την παροχή λύσεων σε πραγματικό χρόνο. Επίσης, τα συστήματα αναζήτησης πληροφοριών χρησιμοποιούν τεχνολογίες NLP για την εξαγωγή σχετικών δεδομένων από μη δομημένα σύνολα πληροφοριών. [6] [2]

2.5.4 Προκλήσεις και Περιορισμοί του NLP

Παρά τη σημαντική πρόοδο, το NLP συνεχίζει να αντιμετωπίζει προκλήσεις. Η κατανόηση της αμφισημίας της γλώσσας, η επεξεργασία πολιτισμικών διαφορών και η αποφυγή μεροληψίας στα δεδομένα αποτελούν βασικά ζητήματα. Τα σύγχρονα γλωσσικά μοντέλα, αν και εξαιρετικά ικανά, εξαρτώνται από την ποιότητα των δεδομένων εκπαίδευσης, γεγονός που μπορεί να επηρεάσει την ακρίβεια και την απόδοσή τους σε μη αντιπροσωπευτικά περιβάλλοντα. [5] [6]

2.5.5 Συμπεράσματα

Η Επεξεργασία Φυσικής Γλώσσας αποτελεί μία από τις πιο κρίσιμες τεχνολογίες της σύγχρονης τεχνητής νοημοσύνης, επιτρέποντας την ενίσχυση της ανθρώπινης-υπολογιστικής αλληλεπίδρασης. Οι εξελίξεις σε μοντέλα μεγάλης κλίμακας, όπως το GPT, έχουν επιτρέψει τη βελτίωση της ακρίβειας και της αποτελεσματικότητας του NLP, καθιστώντας το απαραίτητο εργαλείο για το μέλλον της τεχνολογίας. [2]

2.6 ChatBots

Τα chatbots είναι προγράμματα υπολογιστών σχεδιασμένα να προσομοιώνουν συνομιλίες με ανθρώπους, χρησιμοποιώντας φυσική γλώσσα. Αποτελούν σημαντικό μέρος της σύγχρονης τεχνητής νοημοσύνης και της επεξεργασίας φυσικής γλώσσας (NLP), επιτρέποντας την αυτοματοποίηση επικοινωνιακών διαδικασιών σε διάφορους τομείς, όπως η εξυπηρέτηση πελατών, η εκπαίδευση και η υγειονομική περίθαλψη.

2.6.1 Τεχνολογίες και Προκλήσεις

Η ανάπτυξη των chatbots βασίζεται σε τεχνολογίες όπως η εξαγωγή πληροφορίας, η μηχανική μάθηση και η βαθιά μάθηση. Η κατανόηση της φυσικής γλώσσας (Natural Language Understanding - NLU) είναι κρίσιμη για την αποτελεσματική λειτουργία τους, καθώς επιτρέπει την ανάλυση και ερμηνεία των εισερχόμενων μηνυμάτων. Ωστόσο, προκλήσεις όπως η κατανόηση συμφραζομένων, η διαχείριση αμφισημιών και η παροχή συνεκτικών απαντήσεων παραμένουν σημαντικά ζητήματα στην ανάπτυξη προηγμένων chatbots. [7]

2.6.2 Εφαρμογές και Τάσεις

Τα chatbots έχουν βρει εφαρμογή σε πολλούς τομείς, συμπεριλαμβανομένων των τηλεπικοινωνιών, των τραπεζών, της υγείας και του ηλεκτρονικού εμπορίου. Η ενσωμάτωση τεχνικών βαθιάς μάθησης έχει βελτιώσει την ικανότητά τους να παρέχουν πιο φυσικές και ανθρώπινες αλληλεπιδράσεις. Επιπλέον, η χρήση ενισχυτικής μάθησης επιτρέπει στα chatbots να προσαρμόζονται και να βελτιώνουν τις επιδόσεις τους με την πάροδο του χρόνου, μαθαίνοντας από προηγούμενες αλληλεπιδράσεις. [8]

2.6.3 Σύγκριση Πλατφορμών NLU

Διάφορες πλατφόρμες NLU, όπως οι IBM Watson, Google Dialogflow, Rasa και Microsoft LUIS, προσφέρουν εργαλεία για την ανάπτυξη chatbots. Η επιλογή της κατάλληλης πλατφόρμας εξαρτάται από παράγοντες όπως η ακρίβεια στην κατανόηση της γλώσσας, η ευκολία ενσωμάτωσης και οι συγκεκριμένες ανάγκες της εφαρμογής. Μελέτες έχουν συγκρίνει αυτές τις πλατφόρμες για να βοηθήσουν τους προγραμματιστές στην επιλογή της βέλτιστης λύσης για τις ανάγκες τους. [9]

2.6.4 Προκλήσεις στην Ανάπτυξη Chatbots

Παρά την πρόοδο, η ανάπτυξη chatbots αντιμετωπίζει προκλήσεις, όπως η διαχείριση πολύπλοκων διαλόγων, η κατανόηση συναισθημάτων και η προσαρμογή σε διαφορετικά πολιτισμικά πλαίσια. Η συνεχής έρευνα και ανάπτυξη στον τομέα της τεχνητής νοημοσύνης και του NLP στοχεύει στην αντιμετώπιση αυτών των προκλήσεων, προκειμένου να δημιουργηθούν πιο αποτελεσματικά και ευέλικτα συστήματα. [10]

Συνολικά, τα chatbots αντιπροσωπεύουν ένα δυναμικό εργαλείο στην ψηφιακή εποχή, με συνεχείς βελτιώσεις να ενισχύουν την ικανότητά τους να προσφέρουν ποιοτικές και αποδοτικές υπηρεσίες σε ποικίλα πεδία εφαρμογής.

2.6.5 Διακεκριμένα μοντέλα

Τα διακεκριμένα μοντέλα chatbots, όπως το ChatGPT της OpenAI και το Gemini της Google DeepMind, έχουν διαμορφώσει νέα δεδομένα στην αλληλεπίδραση ανθρώπου-μηχανής, προσφέροντας φυσικότητα, ακρίβεια και δυνατότητες προσαρμογής σε πολλαπλά συμφραζόμενα.

1. ChatGPT

   Το ChatGPT, αναπτυγμένο από την OpenAI, αποτελεί ένα από τα πλέον διακεκριμένα γλωσσικά μοντέλα. Βασίζεται στην αρχιτεκτονική των Transformers, η οποία αξιοποιεί μηχανισμούς αυτοπροσοχής (self-attention) για την κατανόηση και παραγωγή κειμένου. Το ChatGPT εκπαιδεύτηκε σε μεγάλο όγκο δεδομένων, περιλαμβάνοντας βιβλία, άρθρα και συνομιλίες, αποκτώντας έτσι τη δυνατότητα να απαντά σε ερωτήματα, να δημιουργεί συνεκτικό κείμενο και να συμμετέχει σε διαδραστικές συνομιλίες. Ένα από τα κύρια πλεονεκτήματά του είναι η ικανότητά του να συνδυάζει πολυπλοκότητα με ευχρηστία, καθιστώντας το κατάλληλο για χρήση σε εφαρμογές, όπως τα συστήματα υποστήριξης πελατών και η δημιουργία περιεχομένου. [11]

2. Gemini

   Το Gemini, το πολυτροπικό μοντέλο της Google DeepMind, αποτελεί μια καινοτομία στον χώρο της τεχνητής νοημοσύνης. Σε αντίθεση με μοντέλα που επικεντρώνονται αποκλειστικά στο κείμενο, το Gemini μπορεί να επεξεργάζεται πολλαπλούς τύπους δεδομένων, όπως εικόνες, βίντεο και ήχο. Αυτή η πολυτροπική προσέγγιση του επιτρέπει να παρέχει διευρυμένες δυνατότητες κατανόησης και απόκρισης, καθιστώντας το κατάλληλο για προηγμένες εφαρμογές συνομιλίας, πολυμέσων και ανάλυσης δεδομένων. [12]

3. ClaudeAI

   Ένα άλλο αξιόλογο μοντέλο είναι το Claude AI της Anthropic, το οποίο επικεντρώνεται στην ασφάλεια και τη διαφάνεια. Το Claude σχεδιάστηκε ώστε να αποφεύγει τις προκαταλήψεις και να παρέχει εξηγήσιμες απαντήσεις, κάτι που το καθιστά ιδανικό για εφαρμογές όπου η αμεροληψία και η υπευθυνότητα είναι κρίσιμες. [13]

Τα διακεκριμένα μοντέλα chatbots αποτελούν θεμέλιο για τη βελτίωση της αλληλεπίδρασης ανθρώπου-υπολογιστή. Ενσωματώνοντας προηγμένες τεχνολογίες, όπως η αρχιτεκτονική Transformers και οι πολυτροπικές δυνατότητες, τα μοντέλα αυτά επαναπροσδιορίζουν τις δυνατότητες της τεχνητής νοημοσύνης, εξυπηρετώντας ένα ευρύ φάσμα εφαρμογών. Η συνεχής εξέλιξή τους αναμένεται να ενισχύσει περαιτέρω την ανθρώπινη-υπολογιστική συνεργασία.

2.7 Τεχνολογίες Αναγνώρισης και Παραγωγής Ομιλίας

Η αναγνώριση ομιλίας και η σύνθεση ομιλίας αποτελούν βασικές τεχνολογίες στη σύγχρονη έρευνα για την τεχνητή νοημοσύνη, βελτιώνοντας σημαντικά την αλληλεπίδραση μεταξύ ανθρώπων και μηχανών. Το σύστημα STT επιτρέπει στα συστήματα υπολογιστών να κατανοούν και να μεταγράφουν τις ανθρώπινες εκφράσεις σε γραπτό κείμενο, ενώ το σύστημα TTS παράγει ομιλία με φυσικό ήχο από μια δεδομένη εισαγωγή κειμένου. Διεισδύοντας σε όλους σχεδόν τους τομείς, από τους ψηφιακούς βοηθούς και τις πλατφόρμες εξυπηρέτησης πελατών έως το λογισμικό μετάφρασης και τα αυτοκινούμενα οχήματα, οι τεχνολογίες αυτές έχουν επιφέρει μια νέα εποχή.

Ομοίως, η μηχανική μάθηση και οι αλγόριθμοι βαθιάς μάθησης επιφέρουν δραματική βελτίωση της αποτελεσματικότητας, επιτρέποντας τα συστήματα που μπορούν να αντιλαμβάνονται και να συνθέτουν ομιλία με ακρίβεια και φυσικότητα. Παρόλα αυτά, υπάρχουν αρκετές προκλήσεις που υπάρχουν σήμερα -όπως η πολυγλωσσία και η κατανόηση των διαλέκτων- και εγείρονται ηθικές ανησυχίες σχετικά με τον χειρισμό των προσωπικών δεδομένων. Σε αυτό το κεφάλαιο, θα συζητήσουμε τη βάση και τις τεχνολογίες που διέπουν την αναγνώριση και την παραγωγή ομιλίας, παρέχοντας μια πλήρη επισκόπηση των λειτουργιών και των εφαρμογών τους.

2.7.1 Αναγνώριση Ομιλίας (Speech-to-Text - STT)

Το Speech-to-Text (STT) είναι η τεχνολογική εξέλιξη που επιτρέπει τη μετατροπή προφορικού λόγου σε γραπτό κείμενο, βελτιώνοντας την αλληλεπίδραση μεταξύ ανθρώπων και υπολογιστικών συστημάτων. Η διαδικασία αυτή περιλαμβάνει την ανάλυση του ηχητικού σήματος, την αναγνώριση λέξεων και φράσεων μέσα σε αυτό και στη συνέχεια τη μετατροπή τους σε αναπαράσταση κειμένου.

2.7.2 Βασικές αρχές της αναγνώρισης ομιλίας

Η αναγνώριση ομιλίας βασίζεται σε αλγορίθμους μηχανικής μάθησης και βαθιάς μάθησης τελευταίας τεχνολογίας. Τα συστήματα STT εκπαιδεύονται σε μεγάλα σύνολα δεδομένων ομιλίας που τους επιτρέπουν να αναγνωρίζουν μοτίβα και να αντιστοιχούν ακουστικά σήματα με λέξεις. Η έρευνα έχει αποδείξει ότι η εφαρμογή βαθιών νευρωνικών δικτύων έχει βελτιώσει σημαντικά την ακρίβεια των συστημάτων αναγνώρισης ομιλίας. [14]

2.7.3 Προκλήσεις και εξελίξεις

Παρά τις εξελίξεις, η αναγνώριση ομιλίας αντιμετωπίζει ορισμένες προκλήσεις όσον αφορά την πολυγλωσσία, τις διαλεκτικές διαφορές, τον περιβαλλοντικό θόρυβο και τις διαφορές στην προφορά. Η πρόσφατη επιστημονική έρευνα επικεντρώνεται στην ανάπτυξη πολυγλωσσικών μοντέλων που μπορούν να αναγνωρίζουν την ομιλία σε διαφορετικές γλώσσες και διαλέκτους [15]. Επιπλέον, έχουν ενσωματωθεί μεθοδολογίες βαθιάς μάθησης για την κατασκευή πιο ισχυρών και ακριβών συστημάτων αναγνώρισης ομιλίας.

2.7.4 Deepgram: Μια Σύγχρονη Προσέγγιση στην Αναγνώριση Ομιλίας

Το Deepgram είναι μια σύγχρονη εφαρμογή της τεχνολογίας αναγνώρισης ομιλίας που χρησιμοποιεί τεχνικές βαθιάς μάθησης και αλγορίθμους νευρωνικών δικτύων. Το API του Deepgram χρησιμοποιεί εξειδικευμένα μοντέλα που μπορούν να προσαρμοστούν σε διαφορετικές περιπτώσεις χρήσης. Λειτουργεί στο πλαίσιο ενός end-to-end συστήματος, απαλλάσσοντας έτσι από την ανάγκη για παραδοσιακά ακουστικά μοντέλα και συνδυάζοντας τις διαδικασίες εκπαίδευσης και εξόδου σε μία μηχανή.

Σύμφωνα με την τεκμηρίωση του Deepgram, το API παρέχει δυνατότητες όπως:

- •Προσαρμογή μοντέλων για διαφορετικές ρυθμίσεις ομιλίας (π.χ. τηλεφωνικές κλήσεις, podcasts).
- •Υποστήριξη πολλαπλών γλωσσών (όχι για λειτουργία streaming).
- •Η ενσωμάτωση εφαρμογών ροής (streaming) εξασφαλίζει απόδοση σε πραγματικό χρόνο. [16]

Η τεχνολογία αναγνώρισης ομιλίας είναι ένα από τα ταχύτερα αναπτυσσόμενα τμήματα της τεχνητής νοημοσύνης. Η ανάπτυξη και η αξιοποίηση συστημάτων όπως το Deepgram δείχνουν πού οδεύουν στο μέλλον οι εφαρμογές ΣΤΕ. Αυτή η τεχνολογία είναι πολύ σημαντική για να βοηθήσει τους ανθρώπους και τους υπολογιστές να επικοινωνούν. Ωστόσο, εξακολουθεί να είναι πολύ σημαντικό να αντιμετωπιστούν προκλήσεις όπως οι πολιτισμικές διαφορές και οι περιορισμοί στα δεδομένα για να συνεχίσει να αναπτύσσεται ο τομέας.

2.7.5 ElevenLabs

Το ElevenLabs είναι μια πρωτοποριακή πλατφόρμα που ειδικεύεται στις τεχνολογίες Text-to-Speech (TTS), παρέχοντας λύσεις που αξιοποιούν την τεχνητή νοημοσύνη για τη δημιουργία φυσικών, ρεαλιστικών φωνών. Σχεδιάστηκε με στόχο να αναβαθμίσει την ποιότητα της φωνητικής παραγωγής, επιτρέποντας στους χρήστες να δημιουργούν εξατομικευμένες φωνές με υψηλή ακρίβεια και φυσικότητα. Η πλατφόρμα βασίζεται σε προηγμένα νευρωνικά δίκτυα, τα οποία έχουν εκπαιδευτεί σε μεγάλα και ποικίλα σύνολα δεδομένων φωνής. Με τη χρήση αυτών των τεχνολογιών, το ElevenLabs επιτυγχάνει κορυφαία αποτελέσματα στην παραγωγή φωνών, τόσο για εμπορικές εφαρμογές όσο και για προσωπική χρήση, όπως ηχητικά βιβλία, φωνητικοί βοηθοί και προσομοιώσεις ανθρώπινης ομιλίας.

Η τεχνολογία TTS της ElevenLabs ξεχωρίζει για τη δυνατότητα υποστήριξης πολλών γλωσσών και προφορών, προσαρμόζοντας τη φωνή ανάλογα με τις ανάγκες του χρήστη. Η πλατφόρμα επιτρέπει τη χρήση παραμέτρων όπως η ταχύτητα, η χροιά και η συναισθηματική απόδοση, παρέχοντας πλήρη έλεγχο στην προσαρμογή της παραγόμενης φωνής. Ειδικά χαρακτηριστικά, όπως η ικανότητα δημιουργίας πολυγλωσσικών φωνών ή φωνών με συγκεκριμένο συναισθηματικό τόνο, δίνουν τη δυνατότητα εφαρμογής σε σύνθετα έργα, όπως ταινίες, παιχνίδια ή εφαρμογές εκπαίδευσης. Επιπλέον,

το ElevenLabs υποστηρίζει τη ροή δεδομένων μέσω API, γεγονός που διευκολύνει την ενσωμάτωση σε υπάρχοντα συστήματα και εφαρμογές.

Η κύρια καινοτομία της τεχνολογίας TTS του ElevenLabs έγκειται στην προσέγγισή της με βάση τη βαθιά μάθηση (deep learning). Αντί για παραδοσιακά συστήματα που βασίζονται σε προκαθορισμένα δείγματα φωνής, το ElevenLabs χρησιμοποιεί ένα γενικευμένο μοντέλο που μπορεί να προσαρμόσει τη φωνή ανάλογα με τις ανάγκες. Το σύστημα είναι σε θέση να παράγει φωνές που είναι εξαιρετικά κοντά στον ανθρώπινο λόγο, με φυσικές παύσεις και προφορά, αποφεύγοντας τα τεχνητά ή μηχανικά χαρακτηριστικά που είναι συχνά εμφανή σε άλλες TTS πλατφόρμες. Επιπλέον, η χρήση προηγμένων αλγορίθμων κατανόησης συμφραζομένων διασφαλίζει ότι η φωνή που παράγεται είναι προσαρμοσμένη στο πλαίσιο του κειμένου, προσφέροντας καλύτερη κατανόηση και εμπειρία ακρόασης.

Η τεχνολογία TTS του ElevenLabs ξεχωρίζει επίσης για τη χαμηλή καθυστέρηση (latency) κατά τη ροή δεδομένων. Χρησιμοποιώντας API με δυνατότητα συνεχούς ροής (streaming), οι φωνές παράγονται σε πραγματικό χρόνο, κάνοντας την πλατφόρμα ιδανική για εφαρμογές όπως ζωντανές παρουσιάσεις, φωνητική καθοδήγηση και διαδραστικά παιχνίδια. Το σύστημα παρέχει επίσης υψηλή ανθεκτικότητα σε συνθήκες χαμηλού εύρους ζώνης, καθιστώντας το κατάλληλο για χρήση ακόμα και σε απομακρυσμένα ή περιορισμένα δίκτυα.

Οι δυνατότητες και οι ιδιαιτερότητες της τεχνολογίας του ElevenLabs αναδεικνύουν την πλατφόρμα ως έναν κορυφαίο πάροχο TTS υπηρεσιών. Η έμφαση στη φυσικότητα της φωνής, η ευελιξία στις εφαρμογές και η χρήση αιχμής τεχνολογίας καθιστούν το ElevenLabs αναπόσπαστο εργαλείο για τις σύγχρονες ανάγκες παραγωγής φωνητικών δεδομένων. [17]

2.8 OpenAI: Υπηρεσίες και τεχνολογίες

Το OpenAI θεωρείται ένα από τα σημαντικότερα ινστιτούτα στον τομέα της έρευνας και ανάπτυξης της τεχνητής νοημοσύνης, με βασική αποστολή να διασφαλίσει ότι η τεχνητή νοημοσύνη θα ωφελήσει όλη την ανθρωπότητα. Το OpenAI ιδρύθηκε το 2015 και βρίσκεται στην πρώτη γραμμή των τεχνολογιών αιχμής και των καινοτόμων μεθόδων στο τοπίο της τεχνητής νοημοσύνης. Στο OpenAI, δίνεται μεγάλη έμφαση στην έρευνα για την ανάπτυξη προσαρμοστικών, εύρωστων και ασφαλών συστημάτων τεχνητής νοημοσύνης που κατανοούν και αλληλεπιδρούν με τον άνθρωπο με τον πιο φυσικό και πρακτικό τρόπο.

Το OpenAI είναι διάσημο για την εντυπωσιακή ανάπτυξή του στον τομέα των γλωσσικών μοντέλων, ιδίως με την κυκλοφορία των αρχιτεκτονικών Generative Pre-trained Transformer (GPT), οι οποίες έχουν φέρει επανάσταση στον τομέα της επεξεργασίας φυσικής γλώσσας (NLP). Τα μοντέλα αυτά επιδεικνύουν μια εντυπωσιακή ικανότητα να παράγουν συνεκτικό και κατανοητό κείμενο, να συνομιλούν, να απαντούν σε ερωτήσεις και να συνοψίζουν πληροφορίες. Τα GPT-3 και GPT-4 αποτελούν παραδείγματα της τεχνολογικής υπεροχής του OpenAI, προσφέροντας εξαιρετική ακρίβεια

και ευελιξία σε ένα ευρύ φάσμα εφαρμογών στη δημιουργία κειμένων, τη μετάφραση, την εξαγωγή πληροφοριών και τη δημιουργία δημιουργικού περιεχομένου.

Εκτός από την εστίασή του στα γλωσσικά μοντέλα, το OpenAI έχει διευρύνει το πεδίο εφαρμογής του για να συμπεριλάβει τομείς όπως η ρομποτική και η προσομοίωση. Έχει επίσης θέσει σε εφαρμογή προγράμματα σχεδιασμένα για τη βελτίωση της διαφάνειας και της λογοδοσίας των εφαρμογών στην τεχνητή νοημοσύνη. Με αυτόν τον τρόπο, με προσπάθειες όπως ο Κώδικας OpenAI και το DALL-E, ο οργανισμός έχει ενισχύσει την ανθρώπινη δημιουργικότητα και να ενθαρρύνει την υιοθέτηση της τεχνητής νοημοσύνης σε μια πληθώρα πεδίων. [18]

Ένα ιδιαίτερο χαρακτηριστικό της OpenAI είναι η δέσμευσή της για τη διαφάνεια και τη συνεργασία με την επιστημονική κοινότητα. Δημοσιεύει ακαδημαϊκές εργασίες, μοιράζεται εργαλεία και πλατφόρμες ανάπτυξης, και παρέχει ελεύθερη πρόσβαση σε πολλά από τα μοντέλα και τις τεχνολογίες της. Αυτό έχει συμβάλει καθοριστικά στην ευρύτερη διάδοση της AI και στην πρόοδο του πεδίου. Παράλληλα, η OpenAI διατηρεί υψηλά πρότυπα ασφαλείας και δεοντολογίας, λαμβάνοντας υπόψη τις πιθανές προκλήσεις και τις κοινωνικές επιπτώσεις της τεχνητής νοημοσύνης. [19]

Η OpenAI συνεχίζει να διαδραματίζει πρωταγωνιστικό ρόλο στην ενίσχυση της ανθρώπινης γνώσης και δημιουργικότητας μέσω της AI. Με την έρευνά της, την ανάπτυξη καινοτόμων προϊόντων και τη δέσμευση για κοινωνική ευθύνη, η OpenAI θέτει τα θεμέλια για ένα μέλλον όπου η τεχνητή νοημοσύνη θα χρησιμοποιείται για το κοινό καλό.

Το OpenAI έχει συμβάλει σημαντικά στους τομείς της αναγνώρισης ομιλίας και της παραγωγής ομιλίας, αποτελώντας έτσι έναν από τους κύριους παράγοντες στην ανάπτυξη της τεχνητής νοημοσύνης. Μεταξύ των πιο γνωστών επιτευγμάτων του είναι το Whisper, ένα σύστημα αυτόματης αναγνώρισης ομιλίας που έχει εκπαιδευτεί σε ένα τεράστιο σύνολο δεδομένων 680.000 ωρών πολύγλωσσων και πολυδιάστατων πληροφοριών από το διαδίκτυο. Αυτή η εκπαίδευση του Whisper του προσδίδει ισχυρή αντοχή στις αλλαγές της προφοράς, του περιβαλλοντικού θορύβου και του εξειδικευμένου λεξιλογίου, καθιστώντας το ικανό να μεταγράφει πολύ καλά τις ομιλούμενες γλώσσες, μαζί με τη μετάφρασή τους στα αγγλικά. [20]

Το OpenAI έχει αναπτύξει μια σειρά από τεχνολογίες και υπηρεσίες που ενισχύουν την ανθρώπινη-υπολογιστική αλληλεπίδραση, επιλύοντας σύνθετα προβλήματα και προσφέροντας καινοτόμες λύσεις σε πολλούς τομείς. Μία από τις πιο εντυπωσιακές τεχνολογίες είναι το DALL·E, ένα μοντέλο δημιουργίας εικόνων από περιγραφές φυσικής γλώσσας. Το μοντέλο αυτό χρησιμοποιεί την αρχιτεκτονική Transformer, εκπαιδευμένο σε τεράστιους όγκους δεδομένων που συνδυάζουν εικόνες και περιγραφές, επιτρέποντας τη δημιουργία πρωτότυπων και καλλιτεχνικών εικόνων που ανταποκρίνονται με ακρίβεια στις λεκτικές περιγραφές. Το DALL·E έχει βρει εφαρμογή στη διαφήμιση, την καλλιτεχνική δημιουργία, την εκπαίδευση και τη σχεδίαση, ενώ η ικανότητά του να κατανοεί σύνθετα συμφραζόμενα το καθιστά ιδιαίτερα χρήσιμο για καινοτόμες λύσεις. [21]

Ένα άλλο διακεκριμένο μοντέλο είναι το Codex, το οποίο εξειδικεύεται στη σύνθεση και ανάλυση κώδικα. Το Codex εκπαιδεύτηκε σε μεγάλα αποθετήρια δεδομένων κώδικα, όπως το GitHub, και προσφέρει λειτουργίες όπως η αυτόματη δημιουργία, διόρθωση και ερμηνεία κώδικα. Ενσωματωμένο στο GitHub Copilot, το Codex υποστηρίζει τους προγραμματιστές παρέχοντας προτάσεις κώδικα σε πραγματικό χρόνο, μειώνοντας τα λάθη και επιταχύνοντας τη διαδικασία ανάπτυξης λογισμικού. Η ευελιξία του Codex το καθιστά απαραίτητο εργαλείο για ομάδες ανάπτυξης λογισμικού και νέους προγραμματιστές. [22] [23]

Η OpenAI έχει επίσης αναπτύξει το Gym, μια πλατφόρμα που χρησιμοποιείται για την ανάπτυξη και αξιολόγηση αλγορίθμων ενισχυτικής μάθησης (Reinforcement Learning). Το Gym παρέχει ένα ευρύ φάσμα περιβαλλόντων που καλύπτουν από απλά παιχνίδια έως σύνθετες ρομποτικές διεργασίες, διευκολύνοντας την πειραματική έρευνα στον τομέα της μηχανικής μάθησης. Η συμβολή του Gym είναι κρίσιμη για την ανάπτυξη αλγορίθμων που μπορούν να εφαρμοστούν σε ποικίλους τομείς, όπως η ρομποτική και η βελτιστοποίηση διαδικασιών.

Επιπλέον, το OpenAI προσφέρει ένα ολοκληρωμένο API, το οποίο επιτρέπει την πρόσβαση στις δυνατότητες των μοντέλων GPT, Codex και DALL·E. Μέσω του API, προγραμματιστές και επιχειρήσεις μπορούν να ενσωματώσουν την τεχνητή νοημοσύνη στις εφαρμογές τους με απλό και αποδοτικό τρόπο. Το API διευκολύνει την ανάπτυξη εφαρμογών που βασίζονται σε γλωσσικά μοντέλα, δημιουργία περιεχομένου και ανάλυση δεδομένων [24].Οι τεχνολογίες του OpenAI επεκτείνονται περαιτέρω με την πλατφόρμα Spinning Up, η οποία προσφέρει εκπαιδευτικούς πόρους για την ενισχυτική μάθηση, ενισχύοντας τη γνώση στον τομέα για νέους ερευνητές και επαγγελματίες. Με αυτές τις πρωτοποριακές τεχνολογίες, το OpenAI συνεχίζει να διαμορφώνει το μέλλον της τεχνητής νοημοσύνης, προωθώντας την καινοτομία και διευκολύνοντας τη συνεργασία ανθρώπου και μηχανή. [25]

2.9 API

Μια διεπαφή προγραμματισμού εφαρμογών, κοινώς γνωστή ως API, είναι ένα σύνολο πρωτοκόλλων και προτύπων μέσω των οποίων διάφορα συστήματα λογισμικού μπορούν να επικοινωνούν μεταξύ τους. Λειτουργεί ως διεπαφή που επιτρέπει σε μια εφαρμογή να ζητήσει υπηρεσίες ή πληροφορίες από μια άλλη χωρίς να γνωρίζει πώς λειτουργεί εσωτερικά η τελευταία. Με αυτόν τον τρόπο τα API διευκολύνουν την ανάπτυξη λογισμικού, καθώς προωθούν την επαναχρησιμοποίηση του κώδικα και βοηθούν στον συνδυασμό διαφορετικών συστημάτων.

2.9.1 Πώς λειτουργεί το API

Σε γενικές γραμμές, τα API λειτουργούν μέσω προκαθορισμένων κλήσεων ή αιτημάτων από τον πελάτη προς τον διακομιστή. Αυτές μπορεί να είναι αιτήσεις δεδομένων, εκτέλεση λειτουργιών ή πρόσβαση σε συγκεκριμένες υπηρεσίες. Ο διακομιστής επεξεργάζεται το αίτημα και επιστρέφει την κατάλληλη

απάντηση, η οποία μπορεί να είναι δεδομένα, επιβεβαίωση εκτέλεσης ή μήνυμα σφάλματος. Με αυτόν τον τρόπο διαφορετικά λογισμικά μπορούν να συνεργάζονται άψογα μεταξύ τους, ανεξάρτητα από την αρχιτεκτονική ή τη γλώσσα προγραμματισμού τους.

2.9.2 Κατηγορίες API

Υπάρχουν αρκετές ταξινομήσεις API, ανάλογα με τη χρήση τους και τους τομείς εφαρμογών.

- Τα web API επιτρέπουν στους διακομιστές να επικοινωνούν μεταξύ τους μέσω του Διαδικτύου χρησιμοποιώντας πρωτόκολλα όπως το HTTP. Χρησιμοποιούνται κυρίως για την ενσωμάτωση υπηρεσιών ιστού και την ανάπτυξη εφαρμογών ιστού.[26]
- Τα API λειτουργικών συστημάτων παρέχουν διεπαφές μέσω των οποίων οι εφαρμογές αλληλεπιδρούν με το λειτουργικό σύστημα για την εκτέλεση λειτουργιών όπως ο χειρισμός αρχείων και η επικοινωνία με το υλικό.
- Βιβλιοθήκες λογισμικού: Πρόκειται για συλλογές λειτουργιών που οι προγραμματιστές μπορούν να χρησιμοποιούν στις εφαρμογές τους για την εκτέλεση συγκεκριμένων εργασιών, όπως η επεξεργασία εικόνας ή η διαχείριση βάσεων δεδομένων.

2.9.3 Σημασία των API στην ανάπτυξη λογισμικού

Τα APIs αποτελούν ένα πολύ σημαντικό εργαλείο στην ανάπτυξη λογισμικού για τους ακόλουθους λόγους:

- Διευκολύνουν την επαναχρησιμοποίηση του κώδικα, καθώς παρέχουν έτοιμες συναρτήσεις που μπορούν να χρησιμοποιηθούν σε διαφορετικά προγράμματα, μειώνοντας έτσι τον χρόνο ανάπτυξης και τα πιθανά σφάλματα.
- Ενισχύουν τη διαλειτουργικότητα: Επιτρέπουν σε διαφορετικά συστήματα να συνεργάζονται καλά, ανεξάρτητα από τις πλατφόρμες ή ακόμη και τις γλώσσες προγραμματισμού τους.
- Προωθούν την καινοτομία: Ενθαρρύνουν την ανάπτυξη νέων εφαρμογών και υπηρεσιών συνδυάζοντας ήδη υπάρχουσες λειτουργίες και δεδομένα.

2.9.4 Προκλήσεις και βέλτιστες πρακτικές

Παρά τα πλεονεκτήματά τους, τα API παρουσιάζουν επίσης ορισμένες προκλήσεις, όπως:

- Πολυπλοκότητα και αξιοπιστία: Η πολυπλοκότητα ενός API μπορεί να επηρεάσει την υιοθέτησή του από τους προγραμματιστές. Τα απλά και λιγότερο πολύπλοκα API είναι εύκολο να υιοθετηθούν και να ενσωματωθούν. [27]

- Ανακάλυψη λειτουργικότητας: Οι προγραμματιστές πρέπει να ανακαλύψουν ποιες λειτουργίες ενός API είναι σχετικές με το έργο τους και πώς να τις χρησιμοποιήσουν. Η καλή τεκμηρίωση και τα κατάλληλα εργαλεία μπορούν να διευκολύνουν αυτή τη διαδικασία. [28]

- Δοκιμές και επικύρωση: Τα API θα πρέπει να δοκιμάζονται αυτόματα για να διασφαλίζεται ότι λειτουργούν όπως αναμένεται, είναι αξιόπιστα και ασφαλή. Ορισμένες από τις κρίσιμες προκλήσεις για την επίτευξη αυτού του στόχου περιλαμβάνουν την αλληλουχία κλήσεων, τη σύγκριση απρόβλεπτων απαντήσεων και την παράλληλη εκτέλεση κλήσεων. [29]

- Υιοθετούν καλές πρακτικές κατά την ανάπτυξη και τη χρήση των APIs παρέχοντας σαφή και συστηματική τεκμηρίωση, εξασφαλίζοντας την προς τα πίσω συμβατότητα και μελετώντας τους μηχανισμούς ασφαλείας.

2.9.5 Συμπέρασμα

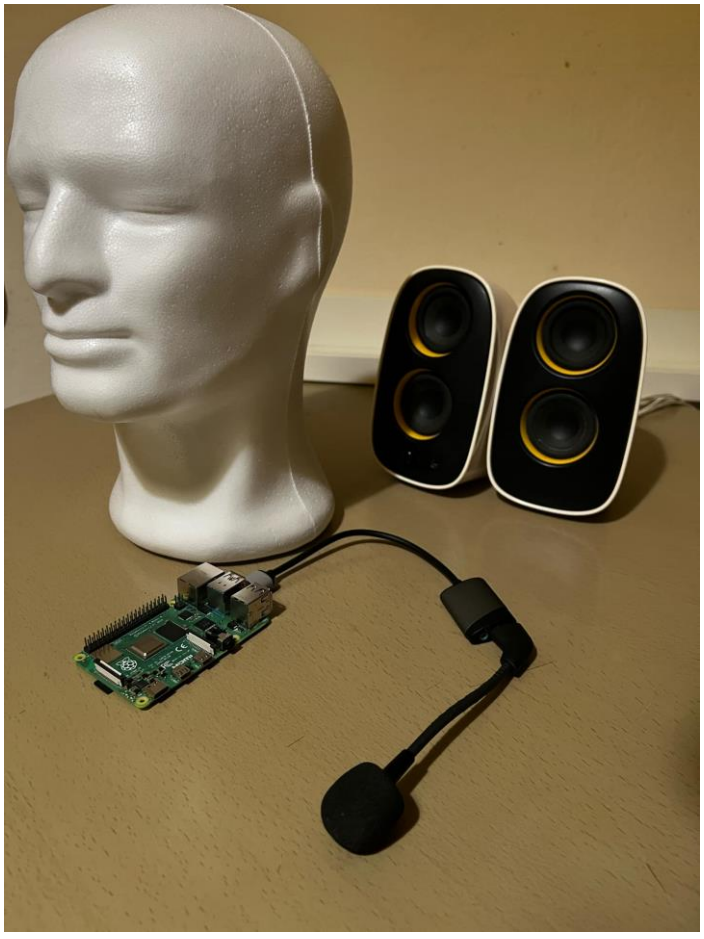
Οι διεπαφές προγραμματισμού εφαρμογών (APIs) αποτελούν βασικές δομές της ανάπτυξης λογισμικού, παρέχοντας αλληλεπίδραση και διαλειτουργικότητα μεταξύ συστημάτων, εφαρμογών και υπηρεσιών. Η σημασία τους δεν περιορίζεται στην παραγωγή νέας τεχνολογίας, αλλά περιλαμβάνει τον εμπλουτισμό υφιστάμενων συστημάτων μέσω της επαναχρησιμοποίησης και ενσωμάτωσης χρήσιμων λειτουργιών. Τα API είναι επίσης σημαντική πηγή εξοικονόμησης χρόνου και πόρων, καθώς οι προγραμματιστές δεν χρειάζεται να υλοποιούν βασικές λειτουργίες, όπως η διαχείριση δεδομένων ή η πρόσβαση σε βάσεις δεδομένων από το μηδέν. Επιπλέον, επιτρέπουν την καινοτομία, διευκολύνοντας τη δημιουργία νέων εφαρμογών βασισμένων σε υπάρχουσες υπηρεσίες. Παράδειγμα είναι συστήματα που χρησιμοποιούν αναγνώριση ομιλίας, επεξεργασία φυσικής γλώσσας και τεχνολογία σύνθεσης ομιλίας, υλοποιούμενα μόνο με API υψηλού επιπέδου, όπως το Deepgram, το OpenAI και το ElevenLabs. Τα API αυξάνουν τη διαλειτουργικότητα, επεκτασιμότητα και αποτελεσματικότητα των συστημάτων. Στο πλαίσιο της παρούσας έρευνας, αποτέλεσαν βασική δομή για τη δημιουργία του συστήματος διαλόγου, συνδυάζοντας τεχνολογίες σε έναν ενιαίο, λειτουργικό χώρο. Η σημασία τους υπερβαίνει αυτή τη χρήση, καθώς είναι κρίσιμη για την ανάπτυξη καινοτόμων λύσεων που προωθούν τη συνδεσιμότητα και λειτουργικότητα.

Κεφάλαιο 3ο: Σχεδιασμός και υλοποίηση συστήματος

3.1 Εισαγωγή

Το Κεφάλαιο 3 εστιάζει στον σχεδιασμό και την υλοποίηση του διαλογικού συστήματος που παρουσιάζεται στη διπλωματική εργασία. Η ανάπτυξη του συστήματος πραγματοποιείται μέσω της ενοποίησης βασικών τεχνολογιών, όπως η αναγνώριση φωνής (Speech-to-Text), η επεξεργασία φυσικής γλώσσας (Natural Language Processing), και η παραγωγή φωνητικής απάντησης (Text-to-Speech). Κάθε υποσύστημα συνδέεται λειτουργικά με τα υπόλοιπα, διασφαλίζοντας μια αρμονική και αποδοτική ροή δεδομένων από την αρχική φωνητική εντολή έως την τελική φωνητική απόκριση. Στόχος του κεφαλαίου είναι να αποσαφηνίσει τον τρόπο με τον οποίο οι επιμέρους τεχνολογίες και ο σχετικός κώδικας υλοποιούνται, ώστε να επιτευχθεί η ολοκληρωμένη λειτουργία του συστήματος.

3.2 Στοιχεία του συστήματος



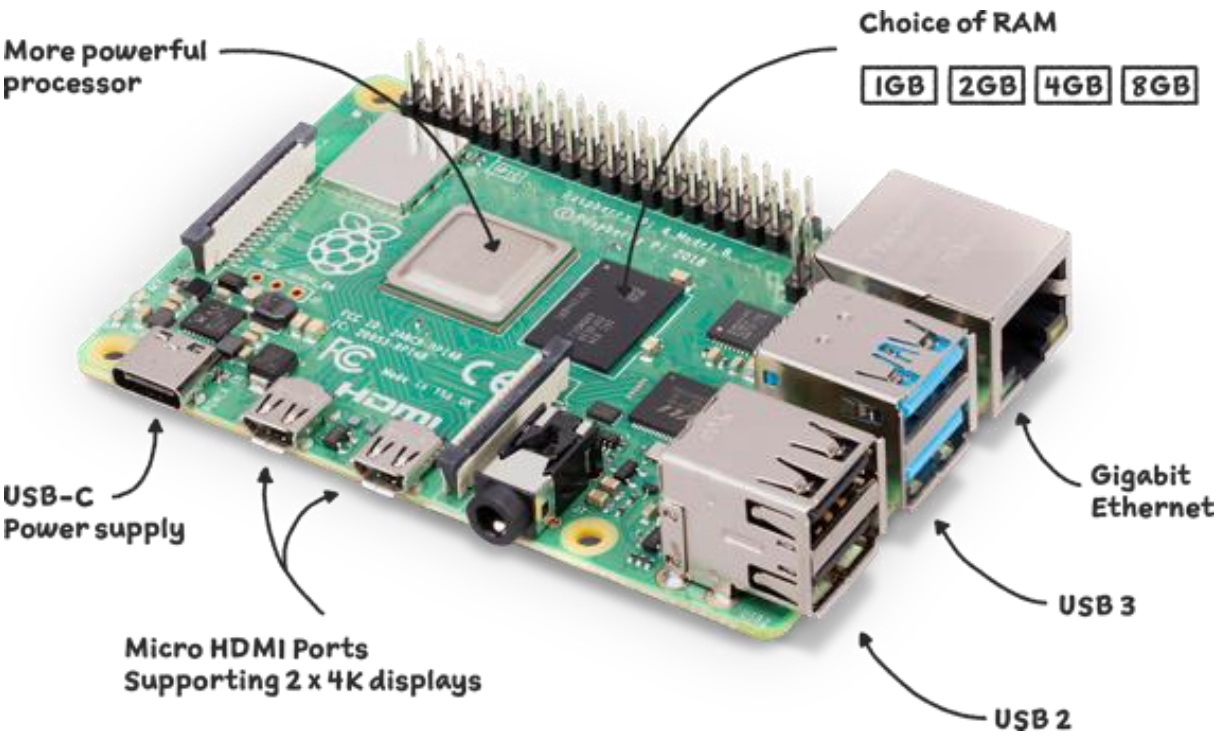
Σχήμα 3.1 Πλήρης διάταξη

Στο σχήμα 3.1 παραπάνω παρουσιάζεται η πλήρης ηλεκτρονική διάταξη που αποτελεί το ολοκληρωμένο σύστημα αλληλεπίδρασης με τον διαλογικό βοηθό. Η χρήση και επιλογή των ηλεκτρονικών αντικειμένων που απεικονίζονται είναι καθαρά προσωπική επιλογή καθώς το ίδιο σύστημα μπορεί να υλοποιηθεί με πολλούς τρόπους αλλά και διαφορετικά ηλεκτρονικά μέρη, για παράδειγμα υπάρχουν πολλές υλοποιήσεις με την χρήση Arduino.

Τα κύρια μέρη της κατασκευής είναι:

- Ανθρώπινο ομοίωμα όπου στεγάζεται η κύρια ηλεκτρονική δομή.
- Ένα Raspberry pi 4 model-B.
- Μια κάρτα ήχου.
- Ένα μικρόφωνο.
- Μια micro SD card.
- Ηχείο.

3.2.1 Raspberry Pi 4



Σχήμα 3.2 Raspberry Pi 4 Model B

[https://www.raspberrypi.com/products/raspberry-pi-4-model-b/]

Το Raspberry Pi 4 Model B αποτελεί έναν μικρό αλλά εξαιρετικά ισχυρό υπολογιστή χαμηλού κόστους σχετικά με έναν κανονικό υπολογιστή, σχεδιασμένο τόσο για εκπαιδευτικούς όσο και προσωπικούς σκοπούς, προσφέροντας έτσι φορητότητα, άνεση αλλά και εξαιρετική επίδοση σε σχέση με τα

προγενέστερα μοντέλα Raspberry Pi. Η παραπάνω εικόνα (σχήμα 3.2) απεικονίζει ένα Raspberry Pi 4 Model B εντοπίζοντας τα κύρια χαρακτηριστικά αλλά και τις εισόδους / εξόδους της πλακέτας.

Μερικά από τα βασικά χαρακτηριστικά του Raspberry Pi 4 Model B είναι :

- Επεξεργαστής (CPU)

  Το BCM2711 είναι το Broadcom chip που χρησιμοποιείται από τα μοντέλα Raspberry Pi 4 Model B, Compute Module 4 και Pi-400. Η αρχιτεκτονική του BCM2711 είναι μια σημαντική αναβάθμιση συγκριτικά με αυτή που έχουν στην κατοχή τους τα SoC σε προηγούμενα μοντέλα Raspberry Pi. Εξακολουθεί να υπάρχει ο τετραπύρηνος σχεδιασμός επεξεργαστή του BCM2711 των 64-bit, ωστόσο χρησιμοποιείται ένας πιο ισχυρός πυρήνας, ο ARM A72 με στόχο την λειτουργία υψηλών επιδόσεων. Οι πυρήνες ARM A72 λειτουργούν με συχνότητες έως και 1,5 GHz καθιστώντας έτσι το μοντέλο Raspberry Pi 4 50% ταχύτερο από το προηγούμενο μοντέλο (Raspberry Pi 3B plus) . Επομένως γίνεται λόγος για έναν επεξεργαστή με πολύ μεγάλη ταχύτητα επεξεργασίας με αποτέλεσμα την μείωση του χρόνου απόκρισης σε σχέση με την προηγούμενη γενιά. Επιπλέον, η αυξημένη ισχύς του τον καθιστά συγκρίσιμο με συστήματα επιπέδου entry-level x86.

- Γραφικά (GPU)

  Διαθέτει δύο θύρες micro HDMI που υποστηρίζουν έως και H.265 δηλαδή αποκωδικοποίηση υλικού 4K στα 60 FPS (4Kp60) με την νέα VideoCore VI 3D να λειτουργεί μέχρι και τα 500MHz με OpenGL ES, graphics 3.0.

- Μνήμη RAM

  Η συσκευή παρέχει μια πληθώρα επιλογών καθώς το μοντέλο εκδίδει συσκευές από 2 έως 8 GB μνήμη, καλύπτοντας έτσι ένα μεγάλο φάσμα εφαρμογών από τις πιο απλές μέχρι εφαρμογές που είναι πιο απαιτητικές σε RAM. Η κατηγοριοποίηση συμβάλλει επίσης και στο κόστος αγοράς καθώς δεν υποχρεώνει τον αγοραστή να επενδύσει σε συσκευή με περίσσεια RAM.

- Συνδεσιμότητα
  - Wi-Fi 802.11ac διπλής ζώνης (2.4 GHz και 5.0GHz) η οποία προσφέρει σταθερή και γρήγορη σύνδεση στο διαδίκτυο.
  - Bluetooth 5.0: Επιτρέπει την ασύρματη σύνδεση με πλήθος περιφερειακών συσκευών.
  - Gigabit (1 GB/s) Ethernet θύρα που παρέχει ενσύρματη σύνδεση μέσω καλωδίου ethernet με πιο σταθερή σύνδεση στο διαδίκτυο με πολύ μεγαλύτερες ταχύτητες για εφαρμογές που απαιτούν μεγαλύτερο εύρος ζώνης.

- Θύρες USB : Διαθέτει 2 θύρες USB 2.0 και δύο θύρες USB 3.0 οι οποίες προσφέρουν μεγαλύτερες ταχύτητες μεταφοράς δεδομένων συγκριτικά με τις USB 2.0.
- Μία θύρα CSI (Camera Serial Interface) για σύνδεση εξειδικευμένων καμερών, ιδανική για εφαρμογές που απαιτούν επεξεργασία εικόνας, όπως συστήματα επιτήρησης
- GPIO: Είσοδοι / Έξοδοι γενικής χρήσης οι οποίοι επιτρέπουν την σύνδεση και την επικοινωνία με διάφορες εξωτερικές συσκευές (πχ. Αισθητήρες).

- Αποθηκευτικός χώρος

  Το Raspberry Pi 4 δεν έχει προ-εγκατεστημένη κάποια μονάδα αποθήκευσης δεδομένων και λειτουργικού συστήματος. Στο πίσω μέρος της πλακέτας υπάρχει μια θήρα για micro SD card που εκεί θα αποθηκεύονται όλα τα δεδομένα καθώς και το λειτουργικό του Raspberry. [30]

3.2.2 Κριτήρια επιλογής του Raspberry Pi 4

Η επιλογή του συγκεκριμένου μοντέλου δηλαδή του Raspberry Pi 4 Model B με 4 GB μνήμη RAM έγινε με γνώμονα τα χαρακτηριστικά που αναφέρονται παραπάνω και το καθιστούν ιδανικό για την υλοποίηση του διαλογικού συστήματος. Πιο συγκεκριμένα :

- Υψηλή υπολογιστική ισχύς: Ο τετραπύρηνος επεξεργαστής ARM Cortex-A72 σε συνδυασμό με την μνήμη RAM των 4GB που επιλέχθηκε για αυτή την υλοποίηση προσφέρουν αρκετή επεξεργαστική ισχύς και μικρούς χρόνους απόκρισης για την διαχείριση εφαρμογών αναγνώρισης φωνής και φυσικής γλώσσας σε πραγματικό χρόνο.
- Σχεδίαση και φορητότητα: Χάρις των μικρών διαστάσεων της πλακέτας αλλά και της ενεργειακής του αποδοτικότητας τον καθιστούν κατάλληλο για την εγκατάστασή του σε ένα μικρό χώρο όπως είναι το ανθρώπινο ομοίωμα.
- Πολλαπλές δυνατότητες συνδεσιμότητας: Παρά την πληθώρα επιλογών στις εισόδους/εξόδους που έχει το συγκεκριμένο μοντέλο για την υλοποίηση του συστήματος, ξεχωρίζουν οι θύρες USB 3.0 αλλά και η Gigabit Ethernet θύρα, καθώς οι μεγάλες ταχύτητες και ο μικρός χρόνος απόκρισης αποτελούν καίρια σημεία της.
- Ευκολία στην ανάπτυξη λογισμικού: Το Raspberry Pi έχει την ικανότητα να υποστηρίζει δημοφιλείς γλώσσες προγραμματισμού, όπως η Python, διαθέτοντας μια ευρεία κοινότητα χρηστών γεγονός που διευκολύνει την ανάπτυξη προγραμμάτων αλλά και στην αντιμετώπιση ενδεχόμενων προβλημάτων.
- Επεκτασιμότητα: Το Raspberry Pi 4 είναι ένα από τα τελευταία μοντέλα της σειράς με αποτέλεσμα να λαμβάνει συνεχώς αναβαθμίσεις στον τρόπο λειτουργίας του. Επίσης, χάρης αυτού η υλοποίηση μπορεί μελλοντικά να αναβαθμιστεί με περισσότερες λειτουργίες , πράγμα που θα αναλυθεί σε επόμενο κεφάλαιο.

Με γνώμονα τα παραπάνω χαρακτηριστικά αλλά και τις απαιτήσεις της κατασκευής το Raspberry Pi 4 Model B 4GB αποτέλεσε την ιδανική επιλογή λαμβάνοντας υπόψιν και το κόστος αγοράς, το οποίο ανέρχεται στα 70€.

3.2.3 Κάρτα ήχου

Σχήμα 3.3 Κάρτα ήχου της Vention
[https://ventiontech.com/products/usb-external-sound-card?srsltid=AfmBOoqp5cVL3WQYgVH9qz-fEuGrhqDOy0MWZa3tg75LqFsFpc7a9LEO]

Η κάρτα ήχου είναι απαραίτητη για την κατασκευή καθώς το Raspberry Pi 4 δεν διαθέτει είσοδο για μικρόφωνο. Επομένως επιλέχθηκε η κάρτα ήχου της Vention. Η εν λόγω κάρτα παράγει υψηλή ποιότητα ήχου 16 bit με εύρος 44100 Hz έως 48000 Hz, παρέχοντας έτσι καθαρό ήχο με ευκρίνεια. Η συμβατότητα της με πολλά λογισμικά , ακόμη και με Raspberry Pi OS αλλά και η φιλική προς τον χρήστη λειτουργικότητα (plug and play) χωρίς να χρειάζεται κάποια επιπλέον εγκατάσταση λογισμικού ή οδηγών εγκατάστασης(drivers) ήταν τα κύρια κριτήρια επιλογής της για να αποφευχθούν περεταίρω επιπλοκές. Κατασκευαστικά είναι απλή και εύχρηστη, έχοντας ένα συμπαγές καλώδιο 15 εκατοστών το οποίο βοηθάει στην δρομολόγηση του μικροφώνου στο επιθυμητό σημείο της κατασκευής με σκοπό την σωστή λειτουργεία του συστήματος εισόδου. Επιπλέον, έχει ενσωματωμένο chip μείωσης θορύβου διασφαλίζοντας ότι ο ήχος που παράγεται στα ηχεία ή συλλέγεται από το μικρόφωνο φιλτράρεται και απομονώνεται από τον θόρυβο, βελτιώνοντας έτσι την εμπειρία του χρήστη. Η σύνδεση της με το Raspberry Pi 4 γίνεται με την χρήση θύρας USB ενώ στο άλλο άκρο εντοπίζονται 2 υποδοχές, μια audio jack 3.5 mm για μικρόφωνο και μία audio jack 3.5 mm για κάποια συσκευή εξόδου, όπως ακουστικά ή ηχεία. Το κόστος για την παραπάνω κάρτα ήχου είναι 7€. [31]

3.2.4 Μικρόφωνο

Σχήμα 3.4 Μικρόφωνο

[https://www.ebay.co.uk/itm/335233912869]

Το μικρόφωνο που χρησιμοποιείται στο σύστημα είναι αποσπώμενο από τα ακουστικά HyperX Cloud II και αποτελεί ένα υψηλής ποιότητας ηλεκτρικό πυκνωτικό μικρόφωνο. Έχει σχεδιαστεί για εφαρμογές που απαιτούν ακριβή καταγραφή φωνής και ελαχιστοποίηση των εξωτερικών παρεμβολών. Το μικρόφωνο διαθέτει μονοκατευθυντική διάταξη (polar pattern) και λειτουργία ακύρωσης θορύβου, καθιστώντας το ιδανικό για περιβάλλοντα με αυξημένα επίπεδα θορύβου.

Οι τεχνικές προδιαγραφές του μικροφώνου είναι οι εξής:

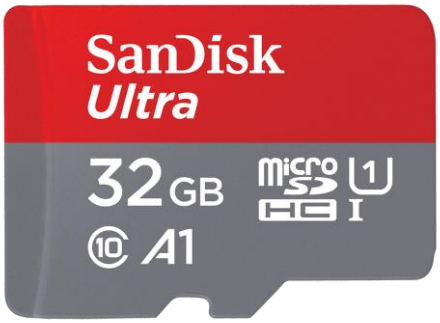
- Στοιχείο: Ηλεκτρικό πυκνωτικό μικρόφωνο.
- Διάταξη καταγραφής (Polar Pattern): Μονοκατευθυντικό, εξασφαλίζοντας την απομόνωση της φωνής του χρήστη από ανεπιθύμητους περιφερειακούς ήχους.
- Ευαισθησία: -42 dBV (1V/Pa στα 1 kHz), διασφαλίζοντας την ακριβή καταγραφή της φωνής ακόμη και σε χαμηλά επίπεδα έντασης.

Η επιλογή του μικροφώνου αυτού βασίζεται στην ικανότητά του να παρέχει καθαρή και ευκρινή είσοδο ήχου, χαρακτηριστικό απαραίτητο για τη λειτουργία του συστήματος αναγνώρισης φωνής. Επιπλέον, η συμβατότητά του με την εξωτερική κάρτα ήχου του συστήματος εξασφαλίζει την απρόσκοπτη ενσωμάτωσή του στην αρχιτεκτονική της κατασκευής. Ως εκ τούτου, το μικρόφωνο HyperX Cloud II

επιλέχθηκε για την αξιοπιστία και την απόδοσή του σε απαιτητικές εφαρμογές φωνητικής αναγνώρισης. [32]

3.2.5 Micro SD Card

Καθώς το Raspberry Pi 4 Model B δεν έχει προ-εγκατεστημένο αποθηκευτικό χώρο, απαιτείται για την ομαλή του λειτουργία μια micro SD card (σχήμα 3.3) στην οποία γίνεται η εγκατάσταση του λειτουργικού συστήματος του Raspberry Pi.



Σχήμα 3.5 SanDisk Ultra microSDHC 32Gb
[https://shop.sandisk.com/el-gr/products/memory-cards/microsd-cards/sandisk-ultra-uhs-i-microsd?sku=SDSQUA4-032G-GN6MA]

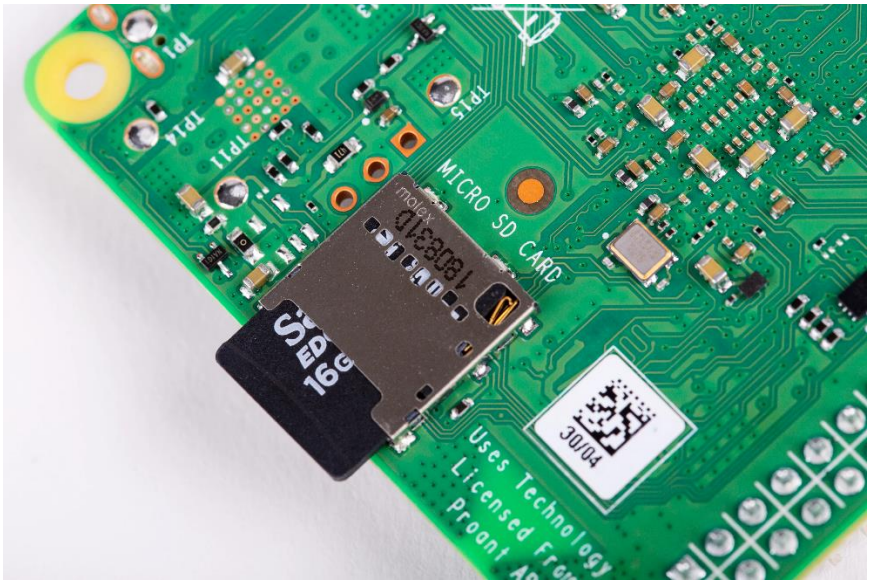
Στην παραπάνω εικόνα παρατηρείται η SanDisk Ultra microSDHC 32Gb με ταχύτητα έως 120 MB/s. Η συγκεκριμένη κάρτα επιλέχθηκε με βασικό κριτήριο την χωρητικότητα των 32 Gb, διότι εκεί εγκαταστάθηκε λειτουργικό σύστημα Raspberry Pi OS. Αξίζει να σημειωθεί ότι το ελάχιστο αποθηκευτικό χώρο που χρειάζεται το Raspberry Pi 4 για να λειτουργεί ομαλά είναι τα 8 Gb. Η micro SD card επιτρέπει την αποθήκευση όλων των απαραίτητων προγραμμάτων και αρχείων του διαλογικού βοηθού, συμπεριλαμβανομένων των βιβλιοθηκών και των εφαρμογών που χρησιμοποιούνται (π.χ., Deepgram SDK, OpenAI API, ElevenLabs SDK). Επίσης η ταχύτητα μεταφοράς δεδομένων της στα 120 MB/s ήταν μια σημαντική προσθήκη για να μειωθεί ο χρόνος της μεταφοράς των δεδομένων από και προς την μονάδα αποθήκευσης, επιτρέποντας στο σύστημα να λειτουργεί ομαλά. Τέλος αξιοπιστία της κατασκευής που εγγυάται η SanDisk συνέβαλε στην αγορά της, καθώς η SanDisk Ultra είναι γνωστή για την ανθεκτικότητά της σε υψηλές θερμοκρασίες, υγρασία, κραδασμούς και μαγνητικά πεδία. Αυτό

την καθιστά ιδανική για συνεχή χρήση σε συστήματα όπως το Raspberry Pi, όπου απαιτείται υψηλή αξιοπιστία και σταθερότητα.

Η επιλογή της σωστής micro SD card είναι κρίσιμη για τη συνολική απόδοση του Raspberry Pi, καθώς λειτουργεί ως κύρια μονάδα αποθήκευσης δεδομένων και του λειτουργικού συστήματος. Μια ταχύτερη κάρτα μειώνει σημαντικά τους χρόνους εκκίνησης του λειτουργικού συστήματος, τη φόρτωση εφαρμογών και τη γενικότερη απόκριση του συστήματος.

Όπως παρατηρείται και από την εικόνα, η κάρτα φέρει πάνω της το λογότυπο A1. Το Raspberry Pi OS προτείνει τη χρήση καρτών κατηγορίας Class 10 ή A1, όπως η SanDisk Ultra που χρησιμοποιείται εδώ, για βέλτιστη απόδοση. Οι κάρτες A1 είναι βελτιστοποιημένες για εφαρμογές, προσφέροντας ταχύτερη τυχαία ανάγνωση και εγγραφή, χαρακτηριστικό απαραίτητο για τη σταθερή λειτουργία του συστήματος.

Η επιλογή μιας κάρτας υψηλής ταχύτητας και αξιοπιστίας, όπως η SanDisk Ultra microSDHC, εξασφαλίζει την ομαλή λειτουργία του λειτουργικού συστήματος και των εφαρμογών, αποτρέποντας καθυστερήσεις ή σφάλματα στη διαδικασία. Η ανθεκτικότητά της και η ικανότητα συνεχούς ανάγνωσης/εγγραφής δεδομένων συμβάλλουν στην απόδοση του συστήματος και στη σταθερή λειτουργία του διαλογικού βοηθού σε πραγματικό χρόνο [33]. Η τοποθέτηση της κάρτας στο Raspberry Pi γίνεται από το πίσω μέρος της πλακέτας όπως φαίνεται στο σχήμα 3.5.



Σχήμα 3.6 Θέση εισόδου SD Card

[https://vilros.com/products/raspberry-pi-4-model-b-1?variant=40809478750302]

3.2.6 Ηχείο

Για την αναπαραγωγή του ήχου, χρησιμοποιήθηκαν τα ηχεία Philips SPA3210/10, τα οποία παρέχουν αξιόπιστη και ποιοτική αναπαραγωγή ήχου, καθιστώντας τα κατάλληλα για τις απαιτήσεις του συστήματος διαλογικού βοηθού. Τα ηχεία συνδέονται με τη χρήση 3.5mm audio jack στην εξωτερική κάρτα ήχου, γεγονός που εξασφαλίζει ανώτερη ποιότητα ήχου σε σχέση με άλλες ασύρματες επιλογές.

Σχήμα 3.7 Ηχεία Philips

[https://www.philips.gr/c-p/SPA3210_10/multimedia-speakers-2.0]

Τεχνικά χαρακτηριστικά

- Ισχύς (RMS): 2 x 2.5 W, με συνολική ισχύ 10 W, ιδανική για καθαρή και δυνατή αναπαραγωγή ήχου σε εσωτερικούς χώρους.
- Έλεγχος Έντασης: Αναλογικός έλεγχος που επιτρέπει την εύκολη προσαρμογή της έντασης του ήχου.
- Βελτίωση Μπάσων: Ενσωματωμένη τεχνολογία για ανάκλαση μπάσων, βελτιώνοντας την ποιότητα των χαμηλών συχνοτήτων.
- Ενσύρματη σύνδεση: Η ενσύρματη σύνδεση συμβάλει στην αξιόπιστη παραγωγή καθαρού ήχου χωρίς διακοπές ή παρεμβολές.

Συμπερασματικά τα κριτήρια επιλογής αυτών των ηχείων έγιναν κυρίως λόγω της ποιότητας ήχου των 10 Watt και της τεχνολογίας βελτίωσης των μπάσων προσφέρουν καθαρή και ισχυρή αναπαραγωγή, απαραίτητη για την κατανόηση των φωνητικών απαντήσεων που παράγονται από το σύστημα TTS.

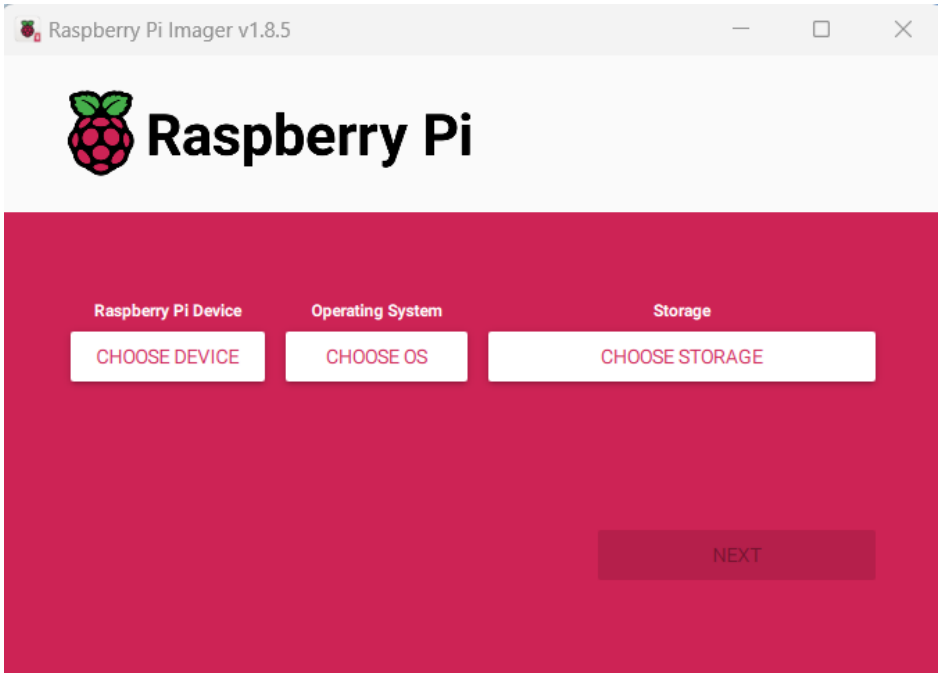
Επιπλέον, χάρις τη θύρα audio jack 3.5 mm, η συμβατότητα με την κάρτα ήχου ήταν ορθή και η επιθυμητή. [34]

Η αρχιτεκτονική του διαλογικού συστήματος που παρουσιάστηκε σε αυτή την ενότητα είναι προφανές ότι αποτελείται από μια σειρά εξειδικευμένων και προσεκτικά επιλεγμένων ηλεκτρονικών στοιχείων, τα οποία λειτουργούν συνδυαστικά για την επίτευξη των στόχων του συστήματος διαλογικού βοηθού με βέλτιστες επιδόσεις. Το Raspberry Pi 4 Model B, με την υψηλή υπολογιστική ισχύ και την πολλαπλή συνδεσιμότητα, αποτελεί τον πυρήνα του συστήματος. Η εξωτερική κάρτα ήχου και το μικρόφωνο HyperX διασφαλίζουν αξιόπιστη καταγραφή και επεξεργασία του ήχου, ενώ η SanDisk Ultra microSDHC εξασφαλίζει αποθηκευτική σταθερότητα και απόδοση. Τα ηχεία Philips SPA3210/10, με την ποιοτική αναπαραγωγή ήχου, ολοκληρώνουν τη λειτουργία του διαλογικού βοηθού, παρέχοντας ένα καθαρό και ευκρινές αποτέλεσμα. Η συνδυαστική λειτουργία αυτών των εξαρτημάτων προσδίδει στο σύστημα ευελιξία, αξιοπιστία και υψηλή απόδοση, ικανοποιώντας τις απαιτήσεις υλοποίησης ενός σύγχρονου διαλογικού βοηθού. Η επιλογή των επιμέρους υλικών έγινε με γνώμονα τη συμβατότητα, την αξιοπιστία και τη δυνατότητα επέκτασης, εξασφαλίζοντας έτσι τη δυνατότητα μελλοντικών βελτιώσεων και προσαρμογών. Συνολικά, το σύστημα που υλοποιήθηκε αποτελεί μία αποτελεσματική και λειτουργική λύση, με προοπτική για περαιτέρω εξέλιξη και βελτιστοποίηση.

3.3 Λογισμικό Συστήματος

Σε αυτή την ενότητα θα παρουσιαστεί το κομμάτι του λογισμικού (Software) που χρησιμοποιήθηκε και εδραιώθηκε πάνω στο hardware της προηγούμενης ενότητας. Καθώς ο βασικότερος πυλώνας αυτής της υλοποίησης στο κομμάτι του hardware είναι το Raspberry το λειτουργικό που χρησιμοποιήθηκε είναι το Raspberry Pi OS, ενώ ο προγραμματισμός του διαλογικού βοηθού έγινε με την χρήση της γλώσσα Python, αλλά και με την βοήθεια άλλων εργαλείων ανάπτυξης όπως οι βιβλιοθήκες.

3.3.1 Raspberry Pi OS



Σχήμα 3.8 Raspberry Pi Imager

Η εγκατάσταση του λειτουργικού συστήματος του Raspberry Pi είναι απλή χωρίς υψηλές απαιτήσεις στο γνωστικό αντικείμενο. Με την χρήση οποιουδήποτε φυλλομετρητή μεταφερόμαστε στην επίσημη ιστοσελίδα του Raspberry Pi όπου εκεί μπορούμε να εγκαταστήσουμε το Software στον υπολογιστή μας. Εφόσον έχουμε εγκατεστημένο το εκτελέσιμο αρχείο στην συσκευή μας, εκτελούμε τον οδηγό εγκατάστασης για το Raspberry Pi Imager. Όταν ολοκληρωθεί η διαδικασία και πλέον έχουμε το Raspberry Pi Imager έτοιμο για εκτέλεση μας μεταφέρει στο παράθυρο του παρακάτω σχήματος.

Οι επιλογές που έγιναν σε αυτή την υλοποίηση είναι:

- Raspberry Pi Device: Raspberry Pi 4
- Operating System: Raspberry Pi OS (64-bit)
- Storage: SDHC CARD

Με την ολοκλήρωση της εγκατάστασης έχουμε ένα πλήρες υπολογιστικό σύστημα και το Raspberry είναι προσβάσιμο στον χρήστη.

3.3.2 Python

Η Python αποτελεί μία από τις πλέον δημοφιλείς και διαδεδομένες γλώσσες προγραμματισμού παγκοσμίως, γνωστή για την απλότητα, την ευελιξία και την πολυλειτουργικότητά της. Δημιουργήθηκε στα τέλη της δεκαετίας του 1980 από τον Guido van Rossum και κυκλοφόρησε επίσημα το 1991. Από τότε, έχει εξελιχθεί σε μία γλώσσα υψηλού επιπέδου που χρησιμοποιείται ευρέως σε πληθώρα

εφαρμογών, από τον ακαδημαϊκό χώρο έως τη βιομηχανία, εξυπηρετώντας τομείς όπως η ανάπτυξη λογισμικού, η επιστημονική έρευνα, η ανάλυση δεδομένων και η τεχνητή νοημοσύνη.

Κύρια Χαρακτηριστικά της Python:

- Γλώσσα Υψηλού Επιπέδου και Αναγνωσιμότητα:

  Η Python ανήκει στις γλώσσες υψηλού επιπέδου, με σύνταξη που είναι εξαιρετικά απλή και κοντά στη φυσική γλώσσα. Αυτό καθιστά τον κώδικα ευανάγνωστο και εύκολα κατανοητό, διευκολύνοντας τους αρχάριους, ενώ παράλληλα επιτρέπει στους έμπειρους προγραμματιστές να αναπτύσσουν πολύπλοκες εφαρμογές με ελάχιστο κώδικα.

- Επεκτασιμότητα μέσω Βιβλιοθηκών:

  Ένα από τα πλέον αξιοσημείωτα χαρακτηριστικά της Python είναι η εκτεταμένη βιβλιοθήκη της, η οποία παρέχει προκατασκευασμένα εργαλεία και πακέτα για ποικίλες εφαρμογές. Ενδεικτικά παραδείγματα είναι:

  - NumPy και Pandas για επιστημονική ανάλυση δεδομένων,
  - Matplotlib και Seaborn για οπτικοποίηση δεδομένων,
  - Django και Flask για ανάπτυξη διαδικτυακών εφαρμογών,
  - TensorFlow και PyTorch για εφαρμογές μηχανικής μάθησης και τεχνητής νοημοσύνης.

- Αντικειμενοστραφής Προγραμματισμός (OOP):

  Η Python υποστηρίζει πλήρως τις αρχές του αντικειμενοστραφούς προγραμματισμού, όπως την κληρονομικότητα, τον πολυμορφισμό και την ενθυλάκωση, επιτρέποντας την οργανωμένη, δομημένη και επεκτάσιμη ανάπτυξη κώδικα.

- Δυναμική Τυποποίηση:

  Η Python είναι δυναμική γλώσσα, γεγονός που σημαίνει ότι οι τύποι δεδομένων προσδιορίζονται κατά τη διάρκεια εκτέλεσης και δεν απαιτείται ρητή δήλωση τύπων από τον προγραμματιστή. Αυτό επιταχύνει τη διαδικασία ανάπτυξης, μειώνοντας το χρόνο σύνταξης και δοκιμής του κώδικα.

- Πλατφόρμα Ανεξαρτησίας:

  Η Python είναι πολυπλατφορμική, καθώς μπορεί να εκτελεστεί σε διαφορετικά λειτουργικά συστήματα (Windows, Linux, macOS) χωρίς να απαιτούνται τροποποιήσεις στον κώδικα.

- Ενεργή Κοινότητα και Υποστήριξη:

  Η Python διαθέτει μία από τις μεγαλύτερες και πιο δραστήριες κοινότητες παγκοσμίως, γεγονός που εξασφαλίζει πληθώρα διαθέσιμων πόρων, τεκμηρίωσης και υποστήριξης για την επίλυση προβλημάτων και την ανάπτυξη καινοτόμων λύσεων.

Πλεονεκτήματα της Python:

- Ευκολία Μάθησης και Χρήσης:
  Η απλότητα της σύνταξής της επιτρέπει σε νέους προγραμματιστές να εστιάσουν στη λύση προβλημάτων αντί για την κατανόηση πολύπλοκων συντακτικών κανόνων.

- Γρήγορη Ανάπτυξη:
  Η συντομία και η αναγνωσιμότητα του κώδικα επιταχύνουν την ανάπτυξη λογισμικού και μειώνουν τα σφάλματα.

- Ευελιξία και Πολλαπλές Εφαρμογές:
  Χρησιμοποιείται σε τομείς όπως η ανάλυση δεδομένων, η ανάπτυξη ιστοσελίδων, οι αυτοματισμοί και η τεχνητή νοημοσύνη.

Μειονεκτήματα της Python:

Παρά τα πλεονεκτήματά της, η Python παρουσιάζει και ορισμένους περιορισμούς:

- Χαμηλότερη Ταχύτητα Εκτέλεσης:
  Ως ερμηνευμένη γλώσσα, η Python είναι πιο αργή σε σύγκριση με γλώσσες χαμηλότερου επιπέδου, όπως η C ή η Java, γεγονός που την καθιστά λιγότερο κατάλληλη για εφαρμογές με υψηλές απαιτήσεις ταχύτητας.

- Αναποτελεσματική Διαχείριση Μνήμης:
  Η δυναμική τυποποίηση και η αυτόματη διαχείριση μνήμης (garbage collection) μπορεί να οδηγήσουν σε αυξημένη κατανάλωση μνήμης γεγονός που αποφεύγεται σε εφαρμογές κινητών συσκευών(mobile applications) . [35]

Συμπερασματικά η Python αποτελεί μία εξαιρετικά ευέλικτη, ευανάγνωστη και ισχυρή γλώσσα προγραμματισμού, κατάλληλη για ένα ευρύ φάσμα εφαρμογών, από την ανάπτυξη διαδικτυακών εφαρμογών έως την ανάλυση δεδομένων και την τεχνητή νοημοσύνη. Παρά τα μειονεκτήματά της σε θέματα ταχύτητας και διαχείρισης μνήμης, η απλότητα και η ισχυρή υποστήριξή της από την κοινότητα την καθιστούν μία από τις κορυφαίες επιλογές στον χώρο του προγραμματισμού και της επιστημονικής έρευνας.

3.3.3 Βιβλιοθήκες

Οι βιβλιοθήκες της Python αποτελούν συλλογές απαραιτήτων λειτουργιών που απαλλάσσουν τον χρήστη από την ανάγκη να αναπτύξει κώδικα από το μηδέν. Η Python διαθέτει μια πληθώρα βιβλιοθηκών που καλύπτουν μεγάλη γκάμα εφαρμογών, καθιστώντας απαραίτητη τη γνώση των καλύτερων εξ αυτών για την ανάπτυξη αποδοτικών και λειτουργικών συστημάτων.

Οι βιβλιοθήκες που χρησιμοποιήθηκαν για το πρόγραμμα σε γλώσσα python είναι:

- Speech Recognition
- Deepgram SDK:

  Η βιβλιοθήκη Deepgram SDK εγκαταστάθηκε για την επικοινωνία με το Deepgram API, το οποίο χρησιμοποιείται για τη μετατροπή ομιλίας σε κείμενο (Speech-to-Text).

  Για να εγκαταστήσουμε την βιβλιοθήκη στο τερματικό γράφουμε την εντολή:

  sudo apt install deepgram-sdk --break-system-packages

  Εφόσον ολοκληρωθεί η εγκατάσταση της βιβλιοθήκης την εισάγουμε στον κώδικα για να μπορεί να χρησιμοποιηθεί με την εντολή:

  import deepgram.

  Ωστόσο, για τον διαλογικό βοηθό, έγινε εξειδικευμένη χρήση της βιβλιοθήκης με την εισαγωγή συγκεκριμένων στοιχείων που είναι απαραίτητα για τη λειτουργία Live Transcription καθώς εκτελείται διαδικασία streaming. Συγκεκριμένα, ο κώδικας περιλαμβάνει:

  from deepgram import (

  LiveTranscriptionEvents,

  LiveOptions,

  Microphone)

- OpenAI:

  Η βιβλιοθήκη OpenAI εγκαταστάθηκε για την επικοινωνία με το OpenAI API, το οποίο χρησιμοποιείται για την επεξεργασία και κατανόηση φυσικής γλώσσας (NLP) για τη δημιουργία απαντήσεων που βασίζονται στην εισαγομένη φωνητική εντολή.

  Η εγκατάσταση της βιβλιοθήκης γίνεται με την εντολή στο τερματικό :

  sudo apt install openai --break-system-packages

  Η εισαγωγή της στον κώδικα γίνεται με την εντολή:

  import openai

- ElevenLabs SDK:

  Η βιβλιοθήκη της ElevenLabs χρησιμοποιείται για τη μετατροπή κείμενου σε φωνή (Text-to-Speech), παρέχοντας υψηλή ποιότητα φωνητικής αναπαραγωγής. Παρέχει καθαρή και φυσική αναπαραγωγή φωνής, η οποία είναι απαραίτητη για τη βέλτιστη εμπειρία του χρήστη.

Η εγκατάσταση γίνεται με την εντολή:

sudo apt install elevenlabs --break-system-packages

Στον κώδικα η χρήση μπορεί να γίνει με την εντολή:

import elevenlabs

Ωστόσο εδώ χρησιμοποιήθηκε ο κώδικας:

from elevenlabs.client import ElevenLabs

from elevenlabs import stream, VoiceSettings

- OS και Time:
  Οι βιβλιοθήκες os και time παρέχουν βασικές λειτουργίες για τη διαχείριση του λειτουργικού συστήματος μέσα από τον κώδικα παραδείγματος χάρη η ανάγνωση ενός εγγράφου και τη μέτρηση καθυστερήσεων στη λειτουργία του κώδικα με την συνάρτηση sleep() η οποία προσθέτει μια καθυστέρηση στην ροή του κώδικα.

- Dotenv:
  Η βιβλιοθήκη dotenv χρησιμοποιείται για τη διαχείριση ευαίσθητων δεδομένων, όπως κλειδιά API και ρυθμίσεις περιβάλλοντος. Η εγκατάσταση της γίνεται με την εντολή:

  sudo apt install python-dotenv --break-system-packages

  ενώ η χρήση της γίνεται με την εντολή:

  from dotenv import load_dotenv()

  Από την βιβλιοθήκη dotenv καλείται η συνάρτηση load_dotenv() η οποία συμβάλει στην διαχείριση μεταβλητών περιβάλλοντος.

Οι παραπάνω βιβλιοθήκες εγκαταστάθηκαν και αξιοποιήθηκαν για την υλοποίηση των βασικών λειτουργιών του συστήματος, προσφέροντας απλοποίηση της ανάπτυξης μέσω προκατασκευασμένων λειτουργιών, βελτιστοποίηση της αποδοτικότητας και ελαχιστοποίηση του χρόνου απόκριση και εξασφάλιση συμβατότητας με το Raspberry Pi και το περιβάλλον Python.

Η επιλογή των βιβλιοθηκών βασίστηκε στην αξιοπιστία και τη διαδεδομένη χρήση τους στην ανάπτυξη συστημάτων φωνητικής επικοινωνίας, επιτυγχάνοντας βέλτιστα αποτελέσματα με ελάχιστο κόστος ανάπτυξης.

Η παράμετρος --break-system-packages επιτρέπει την εγκατάσταση μιας βιβλιοθήκης στο system-wide περιβάλλον καθώς η νέα πολιτική εγκατάστασης των βιβλιοθηκών που προκύπτει λόγω του PEP 668, το οποίο αποτρέπει την απευθείας εγκατάσταση πακέτων μέσω pip. Η χρήση του --break-system-

packages πρέπει να γίνεται μόνο όταν είναι απολύτως απαραίτητο και με προσοχή διότι μπορεί να επηρεάσει την λειτουργία του συστήματος αντικαθιστώντας τα προϋπάρχοντα πακέτα της apt.

3.4 Αρχιτεκτονική Συστήματος

Η αρχιτεκτονική του συστήματος αποτελεί μια συνολική σχεδιαστική προσέγγιση που καθορίζει τη ροή των δεδομένων και τη λειτουργικότητα του διαλογικού βοηθού. Ο σχεδιασμός της αρχιτεκτονικής βασίζεται στην απλότητα, την αποδοτικότητα και την επεκτασιμότητα, με στόχο τη δημιουργία ενός συστήματος που θα λειτουργεί αξιόπιστα, θα ανταποκρίνεται στις απαιτήσεις της εφαρμογής σε πραγματικό χρόνο, αλλά και θα μπορεί μελλοντικά να λάβει περεταίρω βελτίωση με νέες τεχνολογίες τεχνητής νοημοσύνης.

Η συνολική ροή δεδομένων ξεκινά από την καταγραφή της ομιλίας του χρήστη μέσω του μικροφώνου, η οποία εισάγεται στο σύστημα και υποβάλλεται σε επεξεργασία με σκοπό τη μετατροπή της σε κείμενο (Speech-to-Text) με την χρήση του Deepgram API. Στη συνέχεια, το παραγόμενο κείμενο αναλύεται και επεξεργάζεται από το σύστημα επεξεργασίας φυσικής γλώσσας (μέσω του OpenAI API), το οποίο δημιουργεί τη βέλτιστη απάντηση. Τέλος, η απάντηση αυτή μετατρέπεται σε φωνή (Text-to-Speech) και αναπαράγεται στα ηχεία του συστήματος μέσω του ElevenLabs API.

Σκοπός της αρχιτεκτονικής

Η επιλογή αυτής της αρχιτεκτονικής βασίζεται σε τρεις κύριους σκοπούς:

- Απλότητα: Η αρχιτεκτονική θα πρέπει να είναι κατανοητή , επομένως έχει μια βήμα προς βήμα και σαφή ροή πληροφοριών , η οποία μειώνει κάθε πολυπλοκότητα κατά την ανάπτυξη και τη συντήρησή της.
- Αποδοτικότητα: Ο συνδυασμός των τεχνολογιών που εφαρμόζονται διασφαλίζει ότι οι λειτουργίες του συστήματος εκτελούνται με μεγαλύτερη ταχύτητα, γεγονός που περιορίζει τον χρόνο απόκρισης και αναβαθμίζει τη γενική εμπειρία χρήστη.
- Λειτουργικότητα: Το σύστημα είναι σε θέση να εκτελεί θεμελιώδεις λειτουργίες όπως η αναγνώριση φωνής , η κατανόηση φυσικής γλώσσας και η παραγωγή φωνητικών αποκρίσεων με τρόπο που να αποδίδει μια συνεκτική και ευδιάκριτη εκτέλεση.

Η προτεινόμενη αρχιτεκτονική έχει σχεδιαστεί για την επίτευξη της μέγιστης συνεργασίας μεταξύ των εμπλεκόμενων τεχνολογιών, ώστε να υπάρχει ένα αποτελεσματικό και πρακτικό σύστημα επικοινωνίας διαλόγου σε πραγματικό χρόνο.

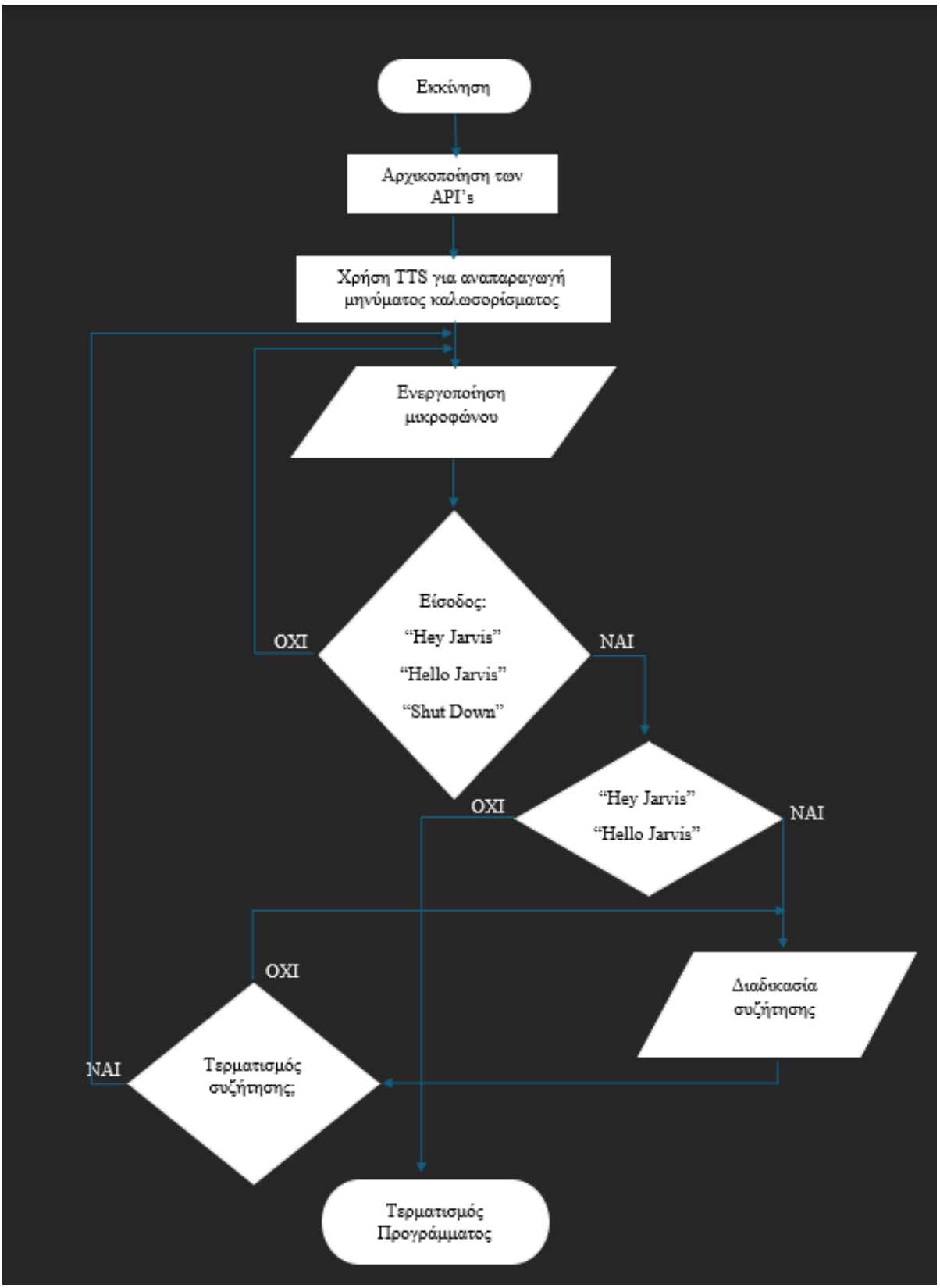
3.5 Διάγραμμα Ροής

Το Διάγραμμα Ροής Δεδομένων (Data Flow Diagram - DFD) απεικονίζει τη συνολική ροή των δεδομένων στο σύστημα, από την εισαγωγή τους μέχρι την τελική έξοδο. Ακολουθεί η περιγραφή και απεικόνιση των βασικών σταδίων:

1. Είσοδος δεδομένων (Input):
    - Ο χρήστης εισάγει την εντολή με την χρήση του μικροφώνου.
    - Η βιβλιοθήκη της Deepgram θέτει τις κατάλληλες ρυθμίσεις και ενεργοποιεί το μικρόφωνο.
2. Μετατροπή της ομιλίας σε κείμενο :
    - Η καταγεγραμμένη φωνή αποστέλλεται στο Deepgram API μέσω της βιβλιοθήκης Deepgram SDK.
    - Το API επιστρέφει την ηχητική είσοδο σε μορφή κειμένου.
3. Επεξεργασία της φυσικής γλώσσας (Natural Language Processing):
    - Το κείμενο αποστέλλεται στο OpenAI API μέσω της βιβλιοθήκης OpenAI.
    - Το API επεξεργάζεται το κείμενο και επιστρέφει την απάντηση σε μορφή κειμένου (Text-to-Text).
4. Μετατροπή του κειμένου σε Ομιλία(Text-to-Speech):
    - Η απάντηση από το OpenAI API μετατρέπεται σε φωνή μέσω του ElevenLabs API.
    - Η διαδικασία πραγματοποιείται με τη χρήση της βιβλιοθήκης ElevenLabs.
5. Έξοδος δεδομένων (Output):
    - Η παραγόμενη φωνή αναπαράγεται μέσω των ηχείων του συστήματος.

Εκτελώντας την παρακάτω ροή τα δεδομένα μετά την εισαγωγή τους εξάγονται ομαλά, γρήγορα και χωρίς σφάλματα καθώς οι τεχνολογίες που επιλέχθηκαν συνεργάζονται ιδανικά μεταξύ τους. Στην συνέχεια γίνεται εκτενής ανάλυση του κώδικα που αναπτύχθηκε για την υλοποίηση του έργου τονίζοντας τα κύρια σημεία του με σκοπό κατανόηση της επιμέρους λειτουργίας κάθε σταδίου της ροής των δεδομένων, από την είσοδο μέχρι και την έξοδο.

Σχήμα 3.9 Μεταφορά δεδομένων



Σχήμα 3.10 Διάγραμμα ροής δεδομένων

Στο παραπάνω διάγραμμα ροής , παρουσιάζεται αναλυτικά η λογική που εφαρμόστηκε για την συγγραφή του πηγαίου κώδικα του συστήματος, με σκοπό να κατανοηθεί πλήρως η λειτουργία και η δομή του. Η επεξήγηση καλύπτει κάθε τμήμα του κώδικα, αναδεικνύοντας τα βασικά σημεία της υλοποίησης, τις χρησιμοποιούμενες βιβλιοθήκες και τα APIs, καθώς και τη ροή δεδομένων μεταξύ αυτών. Το βήμα της διαδικασίας της συζήτησης να αναλυθεί σε ένα μικρότερο διάγραμμα ροής παρακάτω πιο αναλυτικά.

Η ανάλυση χωρίζεται σε υπό-ενότητες, καθεμία από τις οποίες εστιάζει σε συγκεκριμένα τμήματα του κώδικα, όπως η αρχικοποίηση των παραμέτρων, η αναγνώριση ομιλίας (Speech-to-Text), η επεξεργασία φυσικής γλώσσας (Natural Language Processing) και η παραγωγή φωνής (Text-to-Speech). Επιπλέον, περιγράφονται οι κύριες λογικές ροές του προγράμματος, οι μέθοδοι διαχείρισης εντολών του χρήστη, καθώς και οι λειτουργίες τερματισμού και καθαρισμού δεδομένων.

Η παρουσίαση αυτή αποσκοπεί στο να προσφέρει στον αναγνώστη μια ολοκληρωμένη εικόνα της υλοποίησης, ώστε να είναι δυνατή η κατανόηση όχι μόνο της λειτουργίας του συστήματος, αλλά και της λογικής πίσω από τις σχεδιαστικές επιλογές.

3.6 Ανάλυση Κώδικα

Η αρχικοποίηση του προγράμματος αποτελεί το πρώτο και θεμελιώδες βήμα για τη λειτουργία του συστήματος, καθώς εξασφαλίζει τη σωστή ρύθμιση των βιβλιοθηκών και τη διασύνδεση με τα εξωτερικά APIs που χρησιμοποιούνται. Το παρακάτω τμήμα κώδικα επιτελεί αυτή τη λειτουργία:

```
from Deepgram import Speech_to_Text
from OpenAI import ask_chatgpt, conversation_history
import speech_recognition as sr
from dotenv import load_dotenv
import os
import openai
from deepgram import DeepgramClient
from elevenlabs.client import ElevenLabs
from elevenlabs import stream , VoiceSettings
import time

load_dotenv()

DEEPGRAM_API_KEY = os.getenv('deepgram_key')
OPENAI_API_KEY = os.getenv('api_data')
ELEVENLABS_API_KEY = os.getenv('eleven_data')
```

Ως πρώτη ενέργεια είναι η εισαγωγή των απαραίτητων βιβλιοθηκών. Οι βιβλιοθήκες του παραπάνω αποσπάσματος κώδικα έχουν αναφερθεί και εξηγηθεί στην ενότητα Βιβλιοθήκες.

Στην συνέχεια, με την μέθοδο load_dotenv() ενεργοποιείται η δυνατότητα φόρτωσης ευαίσθητων δεδομένων, όπως κλειδιά API, από το αρχείο .env. Το αρχείο αυτό χρησιμοποιείται για τη διαχείριση ευαίσθητων δεδομένων και παραμέτρων ρύθμισης ενός προγράμματος. Στη συγκεκριμένη περίπτωση, χρησιμεύει για την αποθήκευση των κλειδιών API που απαιτούνται για την επικοινωνία με τις εξωτερικές υπηρεσίες (Deepgram, OpenAI, ElevenLabs). Τέλος τα κλειδιά αποθηκεύονται παραμετρικά στις μεταβλητές με τα αντίστοιχα ονόματα.

Εφόσον έχουν εισαχθεί και τα κλειδιά, το επόμενο βήμα είναι η αρχικοποίηση των clients. Ο κώδικας για να υλοποιηθεί αυτό είναι :

```
deepgram: DeepgramClient = DeepgramClient(DEEPGRAM_API_KEY)
openai.api_key = OPENAI_API_KEY
client = ElevenLabs(api_key= ELEVENLABS_API_KEY)
```

Η αρχικοποίηση των clients για τις εξωτερικές υπηρεσίες (Deepgram, OpenAI, ElevenLabs) αποτελεί ένα από τα βασικά βήματα του προγράμματος, καθώς ενεργοποιεί τη δυνατότητα επικοινωνίας με τα APIs. Μέσω αυτής της διαδικασίας, καθίσταται δυνατή η αποστολή αιτημάτων στους διακομιστές (servers) της κάθε τεχνολογίας, ώστε να επεξεργάζονται τα δεδομένα που αποστέλλονται. Ανάλογα με το API, η επικοινωνία μπορεί να πραγματοποιείται μέσω WebSocket ή HTTP πρωτοκόλλου, με στόχο την επιστροφή των επιθυμητών αποτελεσμάτων, όπως κείμενο ή φωνητική έξοδος.

3.6.1 Speech To Text (STT)

Η διαδικασία μετατροπής ομιλίας σε κείμενο (Speech-to-Text, STT) αποτελεί βασικό στοιχείο στη λειτουργία ενός διαλογικού συστήματος, καθώς επιτρέπει την κατανόηση και την ανάλυση των εντολών του χρήστη. Μέσω αυτής της διαδικασίας, η φωνητική είσοδος του χρήστη μετατρέπεται σε κείμενο, το οποίο στη συνέχεια μπορεί να επεξεργαστεί από αλγόριθμους επεξεργασίας φυσικής γλώσσας (Natural Language Processing).

Στο συγκεκριμένο σύστημα, το Deepgram API επιλέχθηκε ως η κύρια τεχνολογία για τη μετατροπή ομιλίας σε κείμενο. Το Deepgram είναι γνωστό για την ακρίβεια και την ταχύτητά του, ενώ υποστηρίζει πολλαπλές γλώσσες και ενσωματώνει τεχνολογίες αναγνώρισης φωνής σε πραγματικό χρόνο μέσω WebSocket. Η ενσωμάτωση αυτής της τεχνολογίας στο σύστημα εξασφαλίζει την αποτελεσματική και άμεση κατανόηση των εντολών του χρήστη, καθιστώντας την εφαρμογή πιο διαδραστική και λειτουργική.

Στο συγκεκριμένο σύστημα, το Deepgram API επιλέχθηκε ως η κύρια τεχνολογία για τη μετατροπή ομιλίας σε κείμενο, λόγω της υψηλής ακρίβειας και της υποστήριξης πολλαπλών γλωσσών. Η

ενσωμάτωσή του μέσω WebSocket διασφαλίζει την ταχύτερη μετάδοση δεδομένων και την άμεση επεξεργασία.

Η ενσωμάτωση του Deepgram API στο σύστημα πραγματοποιείται μέσω του πρωτοκόλλου WebSocket, το οποίο εξασφαλίζει συνεχή ροή δεδομένων μεταξύ του μικροφώνου και του API. Κατά την εκτέλεση, τα ηχητικά δεδομένα αποστέλλονται στο Deepgram, το οποίο αναλαμβάνει την ανάλυση της φωνής και την επιστροφή του αντίστοιχου κειμένου.

Ο τρόπος υλοποίησης αυτής της διαδικασίας περιλαμβάνει τα εξής βήματα:

- Ενεργοποίηση μικροφώνου: Το μικρόφωνο ενεργοποιείται και για την λήψη του σήματος της φωνής του χρήστη.
- Αποστολή του ηχητικού σήματος στο API: Η φωνή αποστέλλεται μέσω του WebSocket στο Deepgram API για φιλτράρισμα και ανάλυση.
- Επιστροφή δεδομένων σε μορφή κειμένου: Το Deepgram API επιστρέφει το αναγνωρισμένο κείμενο, το οποίο στη συνέχεια χρησιμοποιείται από το σύστημα για περαιτέρω επεξεργασία.

Ο κώδικας αρχικοποιεί το Deepgram API μέσω του παρακάτω κώδικα:

```
dg_connection = deepgram.listen.websocket.v("1")
```

Αφού δημιουργηθεί η σύνδεση WebSocket, το επόμενο βήμα είναι η αρχικοποίηση των ρυθμίσεων που καθορίζουν τη λειτουργία του συστήματος Speech-to-Text (STT). Οι ρυθμίσεις υλοποιούνται μέσω του παρακάτω κώδικα:

```
options: LiveOptions = LiveOptions(
            model="nova-2",
            language="en",
            smart_format=True,
            encoding="linear16",
            channels=1,
            sample_rate=16000,
            interim_results=True,
            utterance_end_ms="1000",
            vad_events=True,
            endpointing=300,
        )
```

Βασική λειτουργία του παραπάνω κώδικα είναι η επιλογή του μοντέλου και μερικά από τα πολλά φίλτρα που διαθέτει η τεχνολογία του Deepgram. Πιο συγκεκριμένα :

- Model:

Το Nova-2 αποτελεί μια εξελιγμένη εκδοχή του Nova-1, ενσωματώνοντας βελτιστοποιήσεις στη βασική αρχιτεκτονική του Transformer, προηγμένες τεχνικές επεξεργασίας δεδομένων, καθώς και μια πολυεπίπεδη μεθοδολογία εκπαίδευσης. Οι αναβαθμίσεις αυτές επιτυγχάνουν σημαντική μείωση στο ποσοστό λάθους ανά λέξη (WER) και προσφέρουν ενισχυμένη απόδοση σε τομείς όπως η αναγνώριση οντοτήτων (π.χ. κύρια ονόματα, αλφαριθμητικά δεδομένα), η ακρίβεια στη στίξη, καθώς και η ορθή χρήση κεφαλαίων γραμμάτων.

- Smart Format:
    - Η λειτουργία Έξυπνης Μορφοποίησης (Smart Formatting) της Deepgram παρέχει πρόσθετη επεξεργασία στις μεταγραφές, με στόχο τη βελτιστοποίησή τους για ευκολότερη ανάγνωση από τον άνθρωπο.
    - Οι δυνατότητες της Έξυπνης Μορφοποίησης διαφέρουν ανάλογα με το χρησιμοποιούμενο μοντέλο. Όταν η λειτουργία αυτή είναι ενεργοποιημένη, το Deepgram εφαρμόζει πάντοτε την βέλτιστη διαθέσιμη μορφοποίηση, λαμβάνοντας υπόψη τον συνδυασμό του επιλεγμένου μοντέλου, των παραμέτρων του και της γλώσσας που έχει οριστεί.
- Το API της Deepgram υποστηρίζει πολλές κωδικοποιήσεις για την μεταφορά του ηχητικού σήματος. Το LINEAR16 μεταφέρει δεδομένα PCM WAV 16-bit προσφέροντας έτσι εξαιρετική ποιότητα ήχου χωρίς απώλειες. Η παραπάνω λειτουργία είναι κατάλληλη για εφαρμογές υψηλής ακρίβειας προσφέροντας έτσι στον χρήστη την βέλτιστη εμπειρία.
- Η λειτουργία interim_results είναι ιδιαίτερα χρήσιμη σε εφαρμογές που υλοποιούνται με την διαδικασία streaming. Καθώς γίνεται η δειγματοληψία από το μικρόφωνο το σήμα μεταφέρεται στους διακομιστές της Deepgram μέσω streaming σε ακανόνιστα κομμάτια. Σε ορισμένες περιπτώσεις, η συλλογή του ηχητικού σήματος μπορεί να διακοπεί απότομα, ακόμη και εν μέσω μιας λέξης. Αυτό έχει ως αποτέλεσμα οι προβλέψεις του Deepgram να είναι πιο επιρρεπείς σε σφάλματα, ιδίως όσον αφορά λέξεις που βρίσκονται κοντά στο τέλος της ροής του ήχου. Όταν ενεργοποιείται η λειτουργία ενδιάμεσων αποτελεσμάτων, το Deepgram επιχειρεί να αναγνωρίσει και να μεταγράψει τις λέξεις καθώς εκφωνούνται, αποστέλλοντας τις αρχικές εκτιμήσεις ως ενδιάμεσες μεταγραφές. Καθώς εισέρχεται επιπλέον ηχητικό περιεχόμενο στον διακομιστή, οι μεταγραφές αναθεωρούνται και βελτιώνονται σταδιακά, με αποτέλεσμα την αυξημένη ακρίβεια. Στο τέλος της ηχητικής ροής, το σύστημα παράγει και αποστέλλει μια τελική, αθροιστική μεταγραφή, η οποία αντικατοπτρίζει την πλήρη ανάλυση του ηχητικού σήματος.
- Η λειτουργία UtteranceEnd χρησιμοποιείται για την ανίχνευση του τέλους της ομιλίας κατά τη μεταγραφή ζωντανού ήχου. Συμπληρώνει το Voice Activity Detection (VAD) αναλύοντας τη χρονική τοποθέτηση των λέξεων και τα διαστήματα σιωπής, ώστε να εντοπίζει το τελικό σημείο μιας προφορικής εκφώνησης και να ειδοποιεί τους χρήστες για την ολοκλήρωσή της.

- Η λειτουργία Endpointing του Deepgram εντοπίζει παύσεις που υποδηλώνουν πιθανό τέλος της ομιλίας. Όταν εντοπιστεί τέτοιο σημείο, το σύστημα θεωρεί ότι η μεταγραφή έχει ολοκληρωθεί και δεν απαιτούνται περαιτέρω δεδομένα για τη βελτίωση της πρόβλεψης.

Η αρχική καταχώριση των callbacks γίνεται με τον παρακάτω κώδικα ο οποίος συνδέει τα callbacks με τα αντίστοιχα γεγονότα του Deepgram API:

```
def on_open(self, open, **kwargs):
            print("Connection Open")

        def on_message(self, result, **kwargs):
            nonlocal is_finals
            nonlocal final_text
            sentence = result.channel.alternatives[0].transcript
            if len(sentence) == 0:
                return
            if result.is_final:
                print(f"Message: {result.to_json()}")
                is_finals.append(sentence)

                if result.speech_final:
                    utterance = " ".join(is_finals)
                    final_text = utterance
                    print(f"Speech Final: {utterance}")
                    is_finals = []
                    # Set event to indicate we're done
                    finish_event.set()
            else:
                print(f"Interim Results: {sentence}")

        def on_metadata(self, metadata, **kwargs):
            print(f"Metadata: {metadata}")

        def on_speech_started(self, speech_started, **kwargs):
            print("Speech Started")

        def on_utterance_end(self, utterance_end, **kwargs):
            nonlocal is_finals
            if len(is_finals) > 0:
                utterance = " ".join(is_finals)
                print(f"Utterance End: {utterance}")
                is_finals = []

        def on_close(self, close, **kwargs):
            print("Connection Closed")

        def on_error(self, error, **kwargs):
            print(f"Handled Error: {error}")
```

```
        def on_unhandled(self, unhandled, **kwargs):
            print(f"Unhandled Websocket Message: {unhandled}")

        dg_connection.on(LiveTranscriptionEvents.Open, on_open)
        dg_connection.on(LiveTranscriptionEvents.Transcript, on_message)
        dg_connection.on(LiveTranscriptionEvents.Metadata, on_metadata)
        dg_connection.on(LiveTranscriptionEvents.SpeechStarted, on_speech_started)
        dg_connection.on(LiveTranscriptionEvents.UtteranceEnd, on_utterance_end)
        dg_connection.on(LiveTranscriptionEvents.Close, on_close)
        dg_connection.on(LiveTranscriptionEvents.Error, on_error)
        dg_connection.on(LiveTranscriptionEvents.Unhandled, on_unhandled)
```

Η σύνδεση αυτή διασφαλίζει ότι κάθε callback θα κληθεί όταν συμβεί το αντίστοιχο γεγονός κατά τη διάρκεια της ροής δεδομένων. Παρακάτω παρατίθεται μια εικόνα από την διαδικασία κλήσης των callbacks τα οποία μας δείχνουν την κατάσταση της επεξεργασίας των δεδομένων.

Σχήμα 3.11 Callbacks

Η εικόνα απεικονίζει τη διαδικασία εκτέλεσης του κώδικα για τη λειτουργία του Speech-to-Text μέσω του Deepgram API. Καταγράφεται η επιτυχής έναρξη της σύνδεσης με το WebSocket, η ανίχνευση ομιλίας και η σταδιακή επεξεργασία της φωνητικής εισόδου. Παρουσιάζονται ενδιάμεσα αποτελέσματα αναγνώρισης, τα οποία ενημερώνονται δυναμικά κατά τη διάρκεια της επεξεργασίας, καθώς και τα

τελικά αποτελέσματα της ομιλίας, συμπεριλαμβανομένου του αναγνωρισμένου κειμένου. Παράλληλα, παρατίθενται τα μεταδεδομένα που παρέχουν πληροφορίες για το μοντέλο αναγνώρισης που χρησιμοποιείται, καθώς και για τη συνολική διάρκεια της επεξεργασίας. Τέλος, ολοκληρώνεται η διαδικασία με την ασφαλή αποσύνδεση από το API, επιβεβαιώνοντας την ορθή λειτουργία του συστήματος.

```
if dg_connection.start(options, addons=addons) is False:
            print("Failed to connect to Deepgram")
            return None

        microphone = Microphone(dg_connection.send)
        microphone.start()

        # Χρησιμοποιούμε το Event για να περιμένουμε τον τερματισμό με ασφάλεια
        finish_event.wait()  # Περιμένει μέχρι να οριστεί το event από το τελικό κείμενο

        # Όταν ολοκληρωθεί, κλείνουμε τα πάντα με ασφάλεια
        microphone.finish()
        dg_connection.finish()

        return final_text
    except Exception as e:
        print(f"Could not open socket: {e}")
        return None
```

Το τελικό τμήμα του κώδικα ολοκληρώνει τη διαδικασία αναγνώρισης ομιλίας μέσω του Deepgram API, ενσωματώνοντας κρίσιμα σημεία για την ορθή και αποτελεσματική λειτουργία του συστήματος. Αρχικά, η σύνδεση με το Deepgram API ξεκινά μέσω της μεθόδου dg_connection.start, στην οποία παρέχονται οι ρυθμίσεις και οι πρόσθετες επιλογές που έχουν οριστεί προηγουμένως. Αυτή η μέθοδος επιτρέπει τη σύνδεση με το API, καθιστώντας εφικτή τη μεταφορά δεδομένων ομιλίας για ανάλυση. Σε περίπτωση αποτυχίας σύνδεσης, ο κώδικας διακόπτει τη λειτουργία του και επιστρέφει σχετικό μήνυμα σφάλματος, διασφαλίζοντας ότι ο χρήστης θα ενημερωθεί για το πρόβλημα.

Μετά την επιτυχή σύνδεση, δημιουργείται ένα αντικείμενο Microphone, το οποίο είναι υπεύθυνο για τη μετάδοση των ηχητικών δεδομένων από το μικρόφωνο προς το API. Η ενεργοποίηση του μικροφώνου πραγματοποιείται μέσω της μεθόδου microphone.start, επιτρέποντας τη ροή δεδομένων σε πραγματικό χρόνο. Παράλληλα, ο κώδικας χρησιμοποιεί ένα threading event (finish_event) για τη διαχείριση του συγχρονισμού και την ασφαλή ολοκλήρωση της διαδικασίας. Το event αυτό εξασφαλίζει ότι η εκτέλεση της εφαρμογής αναστέλλεται μέχρι την ολοκλήρωση της ανάλυσης και την επιστροφή του τελικού κειμένου.

Η ολοκλήρωση της διαδικασίας περιλαμβάνει την απενεργοποίηση του μικροφώνου και τον τερματισμό της σύνδεσης με το API, μέσω των μεθόδων microphone.finish και dg_connection.finish, αντίστοιχα. Αυτές οι μέθοδοι αποδεσμεύουν τους χρησιμοποιούμενους πόρους, αποτρέποντας την πιθανότητα σφαλμάτων ή διαρροών μνήμης. Επιπλέον, ο κώδικας περιλαμβάνει μηχανισμούς αντιμετώπισης εξαιρέσεων με τη χρήση της δομής try...except. Αυτή η προσέγγιση επιτρέπει την αποτελεσματική διαχείριση απρόβλεπτων καταστάσεων, διασφαλίζοντας τη σταθερότητα του συστήματος.

Τέλος, η συνάρτηση επιστρέφει το μεταβλητή final_text, η οποία περιέχει το τελικό αποτέλεσμα της αναγνώρισης ομιλίας. Αυτό το αποτέλεσμα είναι απαραίτητο για την περαιτέρω επεξεργασία και την ολοκλήρωση των λειτουργιών του συστήματος. Η συνολική δομή του κώδικα αναδεικνύει τη σημασία της ασφαλούς διαχείρισης πόρων, του συγχρονισμού και της επεξεργασίας εξαιρέσεων, καθιστώντας το σύστημα ανθεκτικό και αποδοτικό σε πραγματικές συνθήκες.

3.6.2 Activation Phrase

Η χρήση της activation phrase είναι ένα κρίσιμο στοιχείο του διαλογικού βοηθού, καθώς επιτρέπει την ενεργοποίηση του συστήματος μέσω συγκεκριμένων φωνητικών εντολών. Αυτή η προσέγγιση παρέχει έναν φυσικό και διαισθητικό τρόπο αλληλεπίδρασης, αποφεύγοντας την ανάγκη χρήσης κουμπιών ή άλλων φυσικών μέσων για την έναρξη της επικοινωνίας. Στο συγκεκριμένο σύστημα, η ενεργοποίηση βασίζεται σε φράσεις όπως "Hey Jarvis" ή "Hello Jarvis", οι οποίες λειτουργούν ως σήμα εκκίνησης για τη λειτουργία αναγνώρισης ομιλίας και επεξεργασίας φυσικής γλώσσας.

Η υλοποίηση της activation phrase ενσωματώνει μηχανισμούς για την ανίχνευση και αναγνώριση αυτών των εντολών σε πραγματικό χρόνο, ενώ παράλληλα εξασφαλίζει την αποτελεσματική διαχείριση πιθανών σφαλμάτων ή θορύβων από το περιβάλλον. Σε αυτή την υποενότητα, παρουσιάζεται η διαδικασία με την οποία ορίζεται, ανιχνεύεται και διαχειρίζεται η activation phrase στο πλαίσιο της συνολικής λειτουργίας του συστήματος. Επιπλέον, αναλύονται τα βασικά τμήματα του κώδικα που υποστηρίζουν τη λειτουργικότητα αυτή, εστιάζοντας στη σημασία της για την εύρυθμη λειτουργία του διαλογικού βοηθού.

Παρακάτω παρατίθεται ο κώδικας με τον οποίο χειρίζεται η διαδικασία της ενεργοποίησης του συστήματος.

```
import speech_recognition as sr
def activation_phrase():
    recognizer = sr.Recognizer()
    microphone = sr.Microphone()

    while True:
        try:
            with microphone as source:
                recognizer.adjust_for_ambient_noise(source, duration=0.4)
```

```
                print("Listening... Please speak.")
                audio = recognizer.listen(source, timeout=5, phrase_time_limit=5)  #
προσθήκη timeouts
                text = recognizer.recognize_google(audio)
                print(f"You said: {text}")
                if "hey jarvis" in text.lower() or "hello jarvis" in text.lower():
                    tts("Hello sir! how can i help you today?",)
                elif "goodbye" in text.lower():
                    tts("Goodbye sir!")
                    break
        except sr.UnknownValueError:
            print("I could not understand you. Please speak again.")
        except sr.RequestError as e:
            tts(f"Could not request results from Google Speech Recognition service; {e}")
        except Exception as e:
            print(f"An unexpected error occurred: {e}")
```

Ο κώδικας παραπάνω είναι αυτός που χρησιμοποιήθηκε στο τελικό πρόγραμμα ωστόσο δεν είναι ο κώδικας του τελικού ολοκληρωμένου , καθώς στον τελικό κώδικα υπάρχουν πρόσθετες γραμμες οι οποίες υλοποιούν άλλες διεργασίες κάνοντας έτσι την επεξήγηση πολύπλοκή. Στο τέλος της διπλωματικής θα υπάρχει ο τελικός κώδικας του έργου.

Ο κώδικας που υλοποιεί τη λειτουργία της activation phrase είναι σχεδιασμένος ώστε να παρακολουθεί συνεχώς την είσοδο του μικροφώνου, ανιχνεύοντας συγκεκριμένες φράσεις όπως "Hey Jarvis" ή "Hello Jarvis". Η παρακάτω ανάλυση εστιάζει στα κύρια σημεία του κώδικα και εξηγεί τη ροή της διαδικασίας.

Γίνεται χρήση της βιβλιοθήκης speech_recognition η οποία προσφέρει εργαλεία για την επεξεργασία φωνητικών δεδομένων, και αξιοποιεί τις κλάσεις Recognizer και Microphone για την αναγνώριση της ομιλίας και τη λήψη ήχου αντίστοιχα.

Αρχικά, το σύστημα ρυθμίζει το κατώφλι θορύβου μέσω της μεθόδου adjust_for_ambient_noise, εξασφαλίζοντας ότι οι αλλαγές στον περιβαλλοντικό θόρυβο δεν επηρεάζουν την ακρίβεια της αναγνώρισης. Στη συνέχεια, η μέθοδος listen καταγράφει φωνητικό δείγμα από τον χρήστη, με παράλληλη εφαρμογή χρονικών περιορισμών (timeout και phrase_time_limit) ώστε να διασφαλίζεται η γρήγορη απόκριση του συστήματος. Οι τιμές στις μεταβλητές καθορίζονται ανάλογα τον τρόπο χρήσης του μοντέλου TTS. Σε αυτήν την υλοποίηση επιλέχθηκε ο συγκεκριμένος κώδικας λόγω της απλότητας του αλλά και της ταχύτητας του καθώς η συγκεκριμένη λειτουργία απλά εντοπίζει μερικές λέξεις. Το ηχητικό δείγμα αναλύεται μέσω της μεθόδου recognize_google, η οποία μετατρέπει τη φωνή σε κείμενο και αποθηκεύει το αποτέλεσμα στη μεταβλητή text.

Ο κώδικας ελέγχει το αναγνωρισμένο κείμενο για την παρουσία των φράσεων "Hey Jarvis" ή "Hello Jarvis". Σε περίπτωση επιτυχούς αναγνώρισης, η συνάρτηση tts καλείται για να εξάγει το ηχητικό μήνυμα "Hello sir! how can I help you today?", υποδεικνύοντας την ενεργοποίηση του συστήματος.

Αντίστοιχα, εάν ο χρήστης πει τη φράση "Goodbye", το σύστημα διακόπτει τη συνομιλία, εξάγοντας το μήνυμα "Goodbye sir!" και τερματίζει τον βρόχο με την εντολή break. Στον κώδικα του κυρίως προγράμματος αυτό το κομμάτι το αναλαμβάνει το Deepgram API με τον εντοπισμό της πρότασης “Thank you” μέσα στην φράση αποχαιρετισμού.

Η υλοποίηση περιλαμβάνει επίσης διαχείριση εξαιρέσεων για την αντιμετώπιση πιθανών προβλημάτων. Αν ο ήχος δεν μπορεί να αναγνωριστεί, το σύστημα διαχειρίζεται το σφάλμα μέσω της εξαίρεσης sr.UnknownValueError, ενώ σε περιπτώσεις προβλημάτων με την υπηρεσία Google Speech Recognition χρησιμοποιείται η εξαίρεση sr.RequestError. Οποιαδήποτε άλλη απρόβλεπτη εξαίρεση καταγράφεται και εμφανίζεται στον χρήστη.

Συνολικά, ο κώδικας προσφέρει έναν αποδοτικό μηχανισμό για την ενεργοποίηση του συστήματος, ενώ η χρήση των κατάλληλων παραμέτρων και η σωστή διαχείριση εξαιρέσεων διασφαλίζουν την ομαλή λειτουργία του.

3.6.3 Επεξεργασία Φυσικής Γλώσσας (NPL)

Το κεφάλαιο αυτό εστιάζει στην ανάλυση και επεξήγηση του κώδικα που αφορά τη λειτουργία του Natural Language Processing (NLP) στο πλαίσιο του διαλογικού συστήματος. Ο πρωταρχικός στόχος του NLP είναι η κατανόηση και επεξεργασία του φυσικού λόγου από τον υπολογιστή, προκειμένου να παρέχει απαντήσεις που είναι ακριβείς και κατανοητές από τον χρήστη.

Στο πλαίσιο της υλοποίησης του συστήματος, η διαδικασία του NLP περιλαμβάνει την αποστολή του κειμένου στο API του OpenAI, την επεξεργασία της ερώτησης, και την επιστροφή της απάντησης. Εδώ θα αναλυθούν τα κύρια μέρη του κώδικα που σχετίζονται με αυτή τη διαδικασία, περιλαμβάνοντας την κλήση του OpenAI API και τη διαχείριση της συνομιλίας με στόχο τη βέλτιστη αλληλεπίδραση με τον χρήστη.

```
import openai
import time


conversation_history = [{"role": "system", "content": "Your name is Jarvis. You are a
really kind, helpful, funny, witty, full of personality and trustful assistant who helps
people solve their problems and answer to their questions.Your creator is Thanasis
Palaiogiannis and he made you for his bachelor's Thesis.Whenever tells you goodbye or thank
you it means that the convesation stops so you greet and close.The answers might be
relatively short and comprehensive "},
               ]
#CHATGPT API GENERATOR
def ask_chatgpt(question, tries=0):
    try:
        conversation_history.append({"role": "user", "content": question})
```

```
        response = openai.chat.completions.create(
            model="gpt-3.5-turbo",
            messages= conversation_history

        )
        answer = response.choices[0].message.content
        conversation_history.append({"role": "assistant", "content": answer})
        return answer
    except openai.APIError as e:
        print(f"Συνέβη ένα λάθος: {e}")
        if isinstance(e, openai.RateLimitError):
            print("Ξεπεράσατε το όριο του API. Περιμένετε για 60 δευτερόλεπτα.")
            time.sleep(60)
            return ask_chatgpt(question, tries=tries+1)
        return None
```

Ο παρεχόμενος κώδικας υλοποιεί τη βασική λειτουργία του Natural Language Processing (NLP) μέσω του OpenAI API, δίνοντας τη δυνατότητα στο σύστημα να επεξεργάζεται ερωτήσεις του χρήστη και να παρέχει απαντήσεις. Η υλοποίηση αυτή βασίζεται στη χρήση του GPT-3.5-turbo, ενός προηγμένου μοντέλου επεξεργασίας φυσικής γλώσσας, για τη δημιουργία συνομιλιών. Η ανάλυση των κύριων σημείων του κώδικα αποκαλύπτει τα εξής:

- Η μεταβλητή conversation_history λειτουργεί ως το αρχείο ιστορικού συνομιλίας, περιλαμβάνοντας τα μηνύματα που ανταλλάσσονται μεταξύ του χρήστη και του συστήματος. Στο αρχικό περιεχόμενό της, περιέχει έναν "ρόλο" συστήματος που ορίζει τον χαρακτήρα του διαλογικού βοηθού, τις ιδιότητες, και τους περιορισμούς του, εξασφαλίζοντας έτσι ότι η συμπεριφορά του βοηθού είναι προσαρμοσμένη στις απαιτήσεις του συστήματος.
- Η συνάρτηση ask_chatgpt(question, tries=0) υλοποιεί την επικοινωνία με το API του OpenAI. Αρχικά, η ερώτηση του χρήστη προστίθεται στο ιστορικό συνομιλίας ως μήνυμα χρήστη. Στη συνέχεια, το σύστημα στέλνει τα μηνύματα της συνομιλίας στο API για επεξεργασία μέσω της μεθόδου openai.chat.completions.create, όπου καθορίζεται το μοντέλο που θα χρησιμοποιηθεί (GPT-3.5-turbo) και τα μηνύματα της συνομιλίας.
- Η απόκριση που λαμβάνεται από το API περιλαμβάνει την απάντηση του βοηθού, η οποία εξάγεται από το πεδίο response.choices[0].message.content. Αυτή η απάντηση προστίθεται στο ιστορικό συνομιλίας ως μήνυμα του βοηθού, διατηρώντας τη συνέχεια της συζήτησης.
- Η συνάρτηση περιλαμβάνει επίσης έναν μηχανισμό διαχείρισης λαθών. Ειδικότερα, σε περίπτωση που παρουσιαστεί σφάλμα του API, όπως η υπέρβαση των ορίων χρήσης (Rate Limit Error), η συνάρτηση περιμένει για 60 δευτερόλεπτα πριν επιχειρήσει ξανά την κλήση

στο API. Ο μηχανισμός αυτός διασφαλίζει την ανθεκτικότητα της εφαρμογής σε ενδεχόμενες δυσλειτουργίες του API.

- Η χρήση αυτής της μεθόδου επιτρέπει στο σύστημα να παρέχει δυναμικές, ευφυείς και φυσικές απαντήσεις στους χρήστες, ενσωματώνοντας παράλληλα μηχανισμούς που διασφαλίζουν τη σταθερότητα και τη συνέχεια της συνομιλίας

Το OpenAI API, και συγκεκριμένα το μοντέλο gpt-3.5-turbo, επιλέχθηκε για την ανάπτυξη του συστήματος φυσικής γλώσσας (Natural Language Processing - NLP) λόγω της υψηλής ακρίβειας, της ευελιξίας και της προσαρμοστικότητάς του. Το συγκεκριμένο μοντέλο προσφέρει δυνατότητα επεξεργασίας και κατανόησης ερωτήσεων με φυσικότητα, ενώ ταυτόχρονα παρέχει συνεκτικές, κατανοητές και περιεκτικές απαντήσεις.

Επιπλέον, το API υποστηρίζει τη χρήση ιστορικού συνομιλίας (conversation history), που επιτρέπει τη δημιουργία ενός διαδραστικού περιβάλλοντος επικοινωνίας, διατηρώντας το πλαίσιο της συνομιλίας για την παροχή πιο σχετικών απαντήσεων. Αυτό είναι ιδιαίτερα κρίσιμο για την εφαρμογή ενός διαλογικού συστήματος, όπως ο ψηφιακός βοηθός "Jarvis".

Η επιλογή του μοντέλου gpt-3.5-turbo έγινε επίσης λόγω της ισορροπίας που προσφέρει μεταξύ απόδοσης και κόστους. Ενώ τα πιο εξελιγμένα μοντέλα, όπως το GPT-4, παρέχουν ελαφρώς βελτιωμένες επιδόσεις, το συγκεκριμένο μοντέλο παραμένει κατάλληλο για εφαρμογές που απαιτούν υψηλή απόδοση με αποδοτική χρήση πόρων.

Τέλος, η υποστήριξη από το OpenAI API για πολλαπλές γλώσσες και η τεκμηριωμένη υποδομή του, το καθιστούν ιδανική επιλογή για εφαρμογές που απαιτούν σταθερότητα, αξιοπιστία και δυνατότητα εύκολης ενσωμάτωσης σε πολυδιάστατα συστήματα.

Συμπερασματικά, η χρήση του OpenAI API στο σύστημα υπογραμμίζει τη σημασία της επιλογής τεχνολογιών που ενσωματώνουν καινοτομία, σταθερότητα και απόδοση. Το API όχι μόνο διευκολύνει τη σύνθετη επεξεργασία φυσικής γλώσσας, αλλά παρέχει και τη βάση για τη δημιουργία ενός λειτουργικού και φιλικού διαλογικού βοηθού. Μέσω της αποτελεσματικής διαχείρισης του ιστορικού συνομιλιών και της ενσωμάτωσης στρατηγικών διαχείρισης λαθών, το OpenAI API αποδεικνύεται αναπόσπαστο μέρος της αρχιτεκτονικής του συστήματος, επιτρέποντας στο Jarvis να παρέχει ακριβείς, χρήσιμες και κατανοητές απαντήσεις στον χρήστη.

3.6.4 Text To Speech (TTS)

Η διαδικασία μετατροπής κειμένου σε ομιλία (Text-to-Speech, TTS) αποτελεί επίσης βασικό μέρος του συστήματος, καθώς επιτρέπει την απόδοση των απαντήσεων του διαλογικού βοηθού με φυσικό και ευδιάκριτο τρόπο. Μέσω της τεχνολογίας TTS, το κείμενο που παράγεται από το σύστημα επεξεργασίας φυσικής γλώσσας (NLP) μετατρέπεται σε ήχο, προσφέροντας μια πιο ρεαλιστική και διαδραστική εμπειρία χρήστη.

Στο συγκεκριμένο σύστημα, χρησιμοποιείται το API της ElevenLabs, το οποίο έχει επιλεγεί λόγω της υψηλής ποιότητας της παραγόμενης φωνής, της υποστήριξης πολλών γλωσσών και της δυνατότητας προσαρμογής παραμέτρων, όπως η ταχύτητα και ο τόνος της ομιλίας. Το ElevenLabs API παρέχει τη δυνατότητα δημιουργίας φωνητικής εξόδου που είναι όχι μόνο κατανοητή, αλλά και ευχάριστη στο άκουσμα, ενισχύοντας την αίσθηση μιας φυσικής συνομιλίας με τον βοηθό.

Στην ενότητα αυτή, θα αναλυθεί η ενσωμάτωση του ElevenLabs API στο σύστημα, καθώς και ο τρόπος υλοποίησης της διαδικασίας TTS. Επιπλέον, θα παρουσιαστούν οι βασικές λειτουργίες και ο κώδικας που χρησιμοποιείται για τη μετατροπή κειμένου σε ομιλία, δίνοντας έμφαση στη ροή δεδομένων και στην προσαρμογή της φωνητικής εξόδου.

```
def text_to_speech(input):
    audio_stream = client.generate(
    text=input,
    stream=True,
    model='eleven_multilingual_v2',
    voice="TX3LPaxmHKxFdv7VOQHJ",
    )
    stream(audio_stream)
```

Ανάλυση κώδικα:

1. Ορισμός της Συνάρτησης Η συνάρτηση text_to_speech δέχεται ως είσοδο το κείμενο που θέλουμε να μετατρέψουμε σε φωνή μέσω της παραμέτρου input. Αυτή η είσοδος προέρχεται συνήθως από τη διαδικασία επεξεργασίας φυσικής γλώσσας (NLP) ή από σταθερά μηνύματα που πρέπει να αναπαραχθούν.
2. Κλήση του API της ElevenLabs Η κλήση του API γίνεται μέσω της μεθόδου client.generate. Η μέθοδος αυτή δημιουργεί ένα φωνητικό stream, το οποίο περιλαμβάνει τα εξής βασικά χαρακτηριστικά:
    - text=input: Το κείμενο που θα μετατραπεί σε φωνή.
    - stream=True: Ενεργοποιεί τη ροή ήχου (streaming), ώστε να επιτρέπει την άμεση αναπαραγωγή καθώς δημιουργείται ο ήχος. Αυτή η επιλογή είναι ιδανική για εφαρμογές πραγματικού χρόνου, όπου η ταχύτητα απόκρισης είναι κρίσιμη.
    - model='eleven_multilingual_v2': Ορίζει το μοντέλο που χρησιμοποιείται για τη μετατροπή. Το συγκεκριμένο μοντέλο υποστηρίζει πολλαπλές γλώσσες, γεγονός που ενισχύει την ευελιξία του συστήματος.

- voice="TX3LPaxmHKxFdv7VOQHJ": Προσδιορίζει την ταυτότητα της φωνής που θα χρησιμοποιηθεί. Οι φωνές μπορούν να είναι είτε προκαθορισμένες είτε εξατομικευμένες, ανάλογα με τις ανάγκες του χρήστη.

3. Εντολή stream(audio_stream): Με την κλήση της εντολής stream(audio_stream), το φωνητικό stream που δημιουργήθηκε από τη μέθοδο generate αναπαράγεται άμεσα μέσω του συστήματος. Αυτό σημαίνει ότι ο χρήστης μπορεί να ακούσει τη φωνή σε πραγματικό χρόνο, χωρίς να χρειάζεται να ολοκληρωθεί πρώτα η δημιουργία του ήχου.

Η χρήση της ροής (streaming) προσφέρει σημαντικά πλεονεκτήματα, όπως:

- Μειωμένη καθυστέρηση στην αναπαραγωγή.
- Βελτιστοποίηση της εμπειρίας χρήστη, ιδίως σε εφαρμογές διαλογικού βοηθού.

Μερικά από τα κρίσιμα σημεία του κώδικα:

- Αποτελεσματικότητα: Η ροή ήχου μέσω της επιλογής stream=True εξασφαλίζει άμεση αναπαραγωγή, καθιστώντας το σύστημα ιδανικό για διαδραστικές εφαρμογές.
- Προσαρμοστικότητα: Η χρήση μοντέλου πολλαπλών γλωσσών και παραμετροποιήσιμων φωνών προσφέρει μεγάλη ευελιξία στη διαχείριση των εξόδων.
- Απλότητα Ενσωμάτωσης: Ο κώδικας είναι αρκετά απλός στη χρήση του, καθιστώντας τη διαδικασία TTS εύκολα υλοποιήσιμη για έναν προγραμματιστή.

3.6.5 Συνολική ροή δεδομένων

Στην ενότητα αυτή, παρουσιάζεται ο πλήρης κώδικας του προγράμματος, ο οποίος συνδυάζει όλες τις επιμέρους λειτουργίες που αναλύθηκαν στις προηγούμενες ενότητες. Ο τελικός κώδικας αποτελεί την ενιαία υλοποίηση του συστήματος, διασφαλίζοντας τη συνεργασία των διαφόρων τεχνολογιών για τη δημιουργία ενός ολοκληρωμένου διαλογικού βοηθού.

Ο στόχος αυτής της ενότητας είναι να καταδείξει πώς όλα τα επιμέρους μέρη, όπως το **Speech-to-Text (STT)**, το **Natural Language Processing (NLP)**, και το **Text-to-Speech (TTS)**, συνδέονται αρμονικά για τη δημιουργία ενός διαλογικού συστήματος που ανταποκρίνεται στις ανάγκες του χρήστη. Επίσης θα αναλυθεί και το διάγραμμα ροής της διαδικασίας της συνομιλίας καθώς πλέον έχουν επεξηγηθεί όλα τα επιμέρους τμήματα κώδικα καθιστώντας την κατανόηση του διαγράμματος πιο εύκολη.

Ο κώδικας ξεκινά με την αρχικοποίηση των API keys και των βιβλιοθηκών που απαιτούνται για την υλοποίηση. Στη συνέχεια, γίνεται χρήση μικροφώνου για τη λήψη φωνητικών εντολών, οι οποίες μετατρέπονται σε κείμενο με τη βοήθεια του Deepgram API. Το παραγόμενο κείμενο επεξεργάζεται

από το OpenAI API για την παραγωγή απαντήσεων, οι οποίες στη συνέχεια μετατρέπονται σε φωνητική έξοδο μέσω του ElevenLabs API.

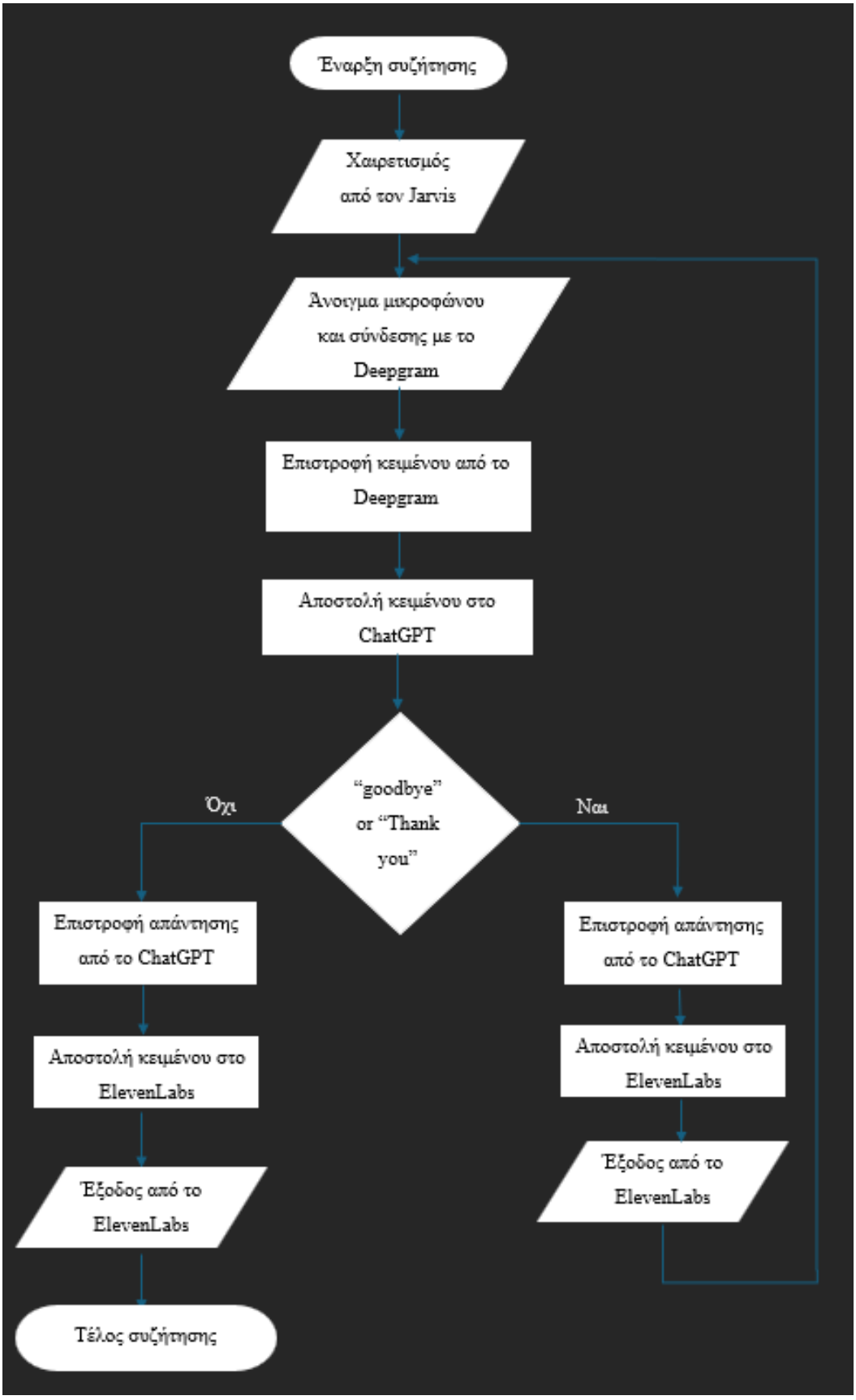
Ο παρακάτω κώδικας παρουσιάζει τη συνολική υλοποίηση:

```
from Deepgram import Speech_to_Text
from OpenAI import ask_chatgpt, conversation_history
import speech_recognition as sr
from dotenv import load_dotenv
import os
import openai
from deepgram import DeepgramClient
from elevenlabs.client import ElevenLabs
from elevenlabs import stream , VoiceSettings
import time

load_dotenv()

DEEPGRAM_API_KEY = os.getenv('deepgram_key')
OPENAI_API_KEY = os.getenv('api_data')
ELEVENLABS_API_KEY = os.getenv('eleven_data')

deepgram: DeepgramClient = DeepgramClient(DEEPGRAM_API_KEY)
openai.api_key = OPENAI_API_KEY
client = ElevenLabs(api_key= ELEVENLABS_API_KEY)

def text_to_speech(input):
    audio_stream = client.generate(
    text=input,
    stream=True,
    model='eleven_multilingual_v2',
    voice="TX3LPaxmHKxFdv7VOQHJ",
    )
    stream(audio_stream)


def main():
    recognizer = sr.Recognizer()
    microphone = sr.Microphone()
    text_to_speech("Welcome to Jarvis project....Please wait!")
    time.sleep(3)
    text_to_speech("You can say 'Hey jarvis!' or 'Hello Jarvis!' and i'll be here for
you!")
    while True:
        try:
            with microphone as source:
                recognizer.adjust_for_ambient_noise(source, duration=0.4)
                print("Listening... Please speak.")
```

```
                audio = recognizer.listen(source, timeout=5, phrase_time_limit=5)  #
προσθήκη timeouts
                text = recognizer.recognize_google(audio)
                print(f"You said: {text}")
                if "hey jarvis" in text.lower() or "hello jarvis" in text.lower():
                    text_to_speech("Hello sir! how can i help you today?")
                    while True:
                        question = Speech_to_Text(deepgram)
                        answer = ask_chatgpt(question)
                        text_to_speech(answer)
                        if "goodbye" in question.lower() or "thank you" in
question.lower():
                          conversation_history = [
                                {
                                    "role": "system",
                                    "content": "Your name is Jarvis. You are a really kind,
helpful, funny, witty, full of personality and trustful assistant who helps people solve
their problems and answer to their questions. Your creator is Thanasis Palaiogiannis and he
made you for his bachelor's Thesis. Whenever he tells you goodbye or thank you, it means
that the conversation stops, so you greet and close. The answers might be relatively short
and comprehensive.In this project you are not a text conversation ai but you are a humanoid
speaker so you can hear what you actually read. "
                                }
                            ]  # Καθαρίζουμε τη μνήμη της συνομιλίας μόνο στο τέλος
                          break
                elif "shut down" in text.lower():
                    break

        except sr.UnknownValueError:
            print("I could not understand you. Please speak again.")
        except sr.RequestError as e:
            print(f"Could not request results from Google Speech Recognition service; {e}")
        except Exception as e:
            print(f"An unexpected error occurred: {e}")
```

Ο ενιαίος κώδικας δείχνει πώς όλες οι επιμέρους λειτουργίες συνδυάζονται για τη δημιουργία ενός λειτουργικού συστήματος. Με την ενσωμάτωση και τη συνεργασία των τεχνολογιών **STT**, **NLP**, και **TTS**, επιτυγχάνεται η αποτελεσματική διαχείριση φωνητικών εντολών, η επεξεργασία φυσικής γλώσσας και η αναπαραγωγή φωνητικών απαντήσεων, δημιουργώντας μια ολοκληρωμένη εμπειρία για τον χρήστη.

3.6.6 Διάγραμμα ροής διαδικασίας συζήτησης



Σχήμα 3.12 Διάγραμμα ροής συζήτησης

Η διαδικασία της συζήτησης ακολουθεί το λογικό πρωτόκολλο επικοινωνίας δυο ανθρώπων. Κατά τον διάλογο ο πρώτος ομιλητής δηλαδή ο χρήστης θέτει την ερώτηση του στον διαλογικό βοηθό (δεύτερο ομιλητή) και αναμένει την απάντηση. Μόλις ο δεύτερος ομιλητής ολοκληρώσει την απάντησή του τότε το μικρόφωνο ανοίγει ξανά και ο χρήστης συνεχίζει τον διάλογο. Παραπάνω δίνεται και το διάγραμμα ροής που χρησιμοποιήθηκε για τον προγραμματισμό.

3.7 Επίλογος

Συνολικά, το Κεφάλαιο 3 καταδεικνύει πώς η υλοποίηση και ο συνδυασμός των επιμέρους τεχνολογιών οδηγούν στη δημιουργία ενός ολοκληρωμένου διαλογικού συστήματος, αναδεικνύοντας την πολύπλευρη συνεργασία των τεχνολογιών Speech-to-Text (STT), Natural Language Processing (NLP) και Text-to-Speech (TTS). Η ολοκλήρωση του συστήματος βασίστηκε στην ενσωμάτωση προσεκτικά επιλεγμένων τεχνολογιών και εργαλείων, όπως το Deepgram, το OpenAI και το ElevenLabs, προσφέροντας υψηλή ακρίβεια, χαμηλό χρόνο απόκρισης και φυσικότητα στις αλληλεπιδράσεις.

Η αρχιτεκτονική του συστήματος σχεδιάστηκε με βασικούς στόχους την αποδοτικότητα, την ευελιξία και την επεκτασιμότητα. Τα επιμέρους υποσυστήματα αναλύθηκαν διεξοδικά, εστιάζοντας στον τρόπο με τον οποίο κάθε στοιχείο συνέβαλε στη συνολική λειτουργία. Το Raspberry Pi 4 αποτέλεσε την καρδιά του συστήματος, προσφέροντας τη δυνατότητα να ενσωματωθούν με επιτυχία τα υπόλοιπα υλικά, όπως το μικρόφωνο, τα ηχεία και η εξωτερική κάρτα ήχου, επιτρέποντας την ακριβή καταγραφή, ανάλυση και αναπαραγωγή δεδομένων.

Επιπλέον, ο συνδυασμός των τεχνολογιών αυτών κατέδειξε την πρακτική εφαρμογή της τεχνητής νοημοσύνης σε πραγματικά σενάρια. Η χρήση προηγμένων αλγορίθμων για την κατανόηση και την επεξεργασία της φυσικής γλώσσας κατέστησε δυνατή τη δημιουργία ενός συστήματος που λειτουργεί ομαλά σε πραγματικό χρόνο. Η εμπειρία του χρήστη ενισχύθηκε σημαντικά μέσω της φυσικής αναπαραγωγής φωνής, της ευχέρειας στη διάδραση και της δυνατότητας σύνθετων λειτουργιών, όπως η διαχείριση ερωτήσεων και απαντήσεων με ακρίβεια.

Η ενότητα αυτή υπογραμμίζει τη σημασία της συνεργασίας πολλαπλών τεχνολογιών για την επίτευξη ενός λειτουργικού και χρήσιμου συστήματος. Το έργο αυτό θέτει τις βάσεις για περαιτέρω ανάπτυξη, επιτρέποντας τη συμπερίληψη νέων χαρακτηριστικών, όπως η υποστήριξη περισσότερων γλωσσών, η ενσωμάτωση ρομποτικής όρασης και η επέκταση της λειτουργικότητας σε περιβάλλοντα εκτός διαδικτύου.

Η επιτυχής ολοκλήρωση του διαλογικού συστήματος καταδεικνύει τις δυνατότητες των τεχνολογιών αυτών να συνδυαστούν σε ένα ενιαίο και λειτουργικό πλαίσιο, ανοίγοντας τον δρόμο για την ανάπτυξη εφαρμογών που εξυπηρετούν εκπαιδευτικούς, επαγγελματικούς και κοινωνικούς σκοπούς.

Κεφάλαιο 4ο: Αποτελέσματα και Αξιολόγηση

4.1 Αξιολόγηση Συστήματος

Η αξιολόγηση του συστήματος πραγματοποιήθηκε με βάση τρεις βασικούς δείκτες: τον χρόνο απόκρισης, την ακρίβεια των τεχνολογιών Speech-to-Text (STT), Text-to-Speech (TTS), και τη λειτουργία του ChatGPT για κατανόηση και παραγωγή φυσικής γλώσσας, καθώς και τη συνολική απόδοση του συστήματος κατά τη χρήση.

4.1.1 Χρόνος Απόκρισης

Η μέτρηση του χρόνου απόκρισης έγινε σε πραγματικές συνθήκες λειτουργίας του συστήματος. Καταγράφηκαν τα εξής:

- Χρόνος αναγνώρισης ομιλίας (STT): Κατά μέσο όρο, το Deepgram API επέστρεφε τα αποτελέσματα σε 1-1,5 δευτερόλεπτα, εξασφαλίζοντας άμεση επεξεργασία της εισερχόμενης ομιλίας.
- Χρόνος παραγωγής ομιλίας (TTS): Η χρήση του ElevenLabs API παρουσίασε εξαιρετική ταχύτητα, με μέσο χρόνο απόκρισης περίπου 1 με 1,5 δευτερόλεπτα.
- Χρόνος κατανόησης και παραγωγής από το ChatGPT: Η διαδικασία επεξεργασίας των δεδομένων από το ChatGPT (μέσω του OpenAI API) είχε μέσο χρόνο απόκρισης περίπου 2 δευτερόλεπτα. Ο χρόνος αυτός περιλάμβανε την κατανόηση της εισόδου και τη δημιουργία της απάντησης. Ο χρόνος απόκρισης σαφώς επηρεάζεται και από το πόσο μακροσκελής είναι η απάντηση.
- Συνολικός χρόνος απόκρισης του συστήματος: Από τη στιγμή που ο χρήστης ενεργοποιούσε το σύστημα μέχρι να παραχθεί η απάντηση, ο συνολικός χρόνος κυμαινόταν μεταξύ 3-4 δευτερολέπτων, περιλαμβάνοντας όλα τα στάδια επεξεργασίας.

4.1.2 Ακρίβεια STT/TTS/OpenAI

Η ακρίβεια των τεχνολογιών αξιολογήθηκε με τη χρήση δοκιμαστικών σεναρίων και φράσεων:

- Speech-to-Text (STT): Το Deepgram API εμφάνισε ακρίβεια 98% σε περιβάλλοντα με χαμηλό θόρυβο. Για περιβάλλοντα με μέτριο θόρυβο, η ακρίβεια μειώθηκε στο 90%-95%.
- Text-to-Speech (TTS): Η ποιότητα της παραγόμενης ομιλίας από το ElevenLabs χαρακτηρίστηκε άριστη, με φυσικότητα, σωστή προσωδία και σαφή προφορά. Επίσης δίνεται σημαντική έμφαση στα σημεία στίξης προσφέροντας την αίσθηση της εναλλαγής συναισθημάτων κατά τον λόγο.
- ChatGPT: Το σύστημα εμφάνισε ακρίβεια και συνέπεια στις απαντήσεις του. Ειδικότερα:
    - Σύντομες και απλές ερωτήσεις: 100% επιτυχία στην κατανόηση.

- Σύνθετες ερωτήσεις: Περίπου 85%-90% ακρίβεια, με μικρές αποκλίσεις σε πολύπλοκες έννοιες.

4.1.3 Συνολική Απόδοση

Το σύστημα λειτούργησε ομαλά, ανταποκρινόμενο στις απαιτήσεις του χρήστη. Κατά τη διάρκεια των δοκιμών:

- Δεν καταγράφηκαν αποτυχίες επικοινωνίας με τα API.
- Η αξιοπιστία του ChatGPT ενίσχυσε τη φυσικότητα της αλληλεπίδρασης, παρέχοντας απαντήσεις που ανταποκρίνονταν στο περιεχόμενο της εισόδου.
- Η συνδυαστική χρήση των τεχνολογιών εξασφάλισε υψηλά επίπεδα ικανοποίησης από τον χρήστη.

4.2 Δυσκολίες και περιορισμοί

Κατά την ανάπτυξη του συστήματος, προέκυψαν διάφορες προκλήσεις που απαιτούσαν ειδική αντιμετώπιση:

Δυσκολίες

Θόρυβος περιβάλλοντος:

- Ο αυξημένος θόρυβος επηρέαζε την ακρίβεια του STT.

Λύση: Χρήση noise cancellation στο μικρόφωνο και προκαταρκτική επεξεργασία του ήχου.

Ενσωμάτωση πολλαπλών API:

- Η διαχείριση της επικοινωνίας μεταξύ των Deepgram, ElevenLabs, και OpenAI APIs απαιτούσε βελτιστοποίηση.
  Λύση: Εφαρμογή ασύγχρονου προγραμματισμού για ταχύτερη απόκριση.
- Πολυπλοκότητα κατανόησης:
  Σε ορισμένες περιπτώσεις, το ChatGPT έδωσε απαντήσεις που δεν ανταποκρίνονταν πλήρως στην πρόθεση του χρήστη.

  Λύση: Προσαρμογή των ερωτήσεων σε πιο συγκεκριμένη μορφή.

Περιορισμοί

Εξάρτηση από σύνδεση στο διαδίκτυο: Το σύστημα βασίζεται στη συνεχή διαδικτυακή σύνδεση για την επικοινωνία με τα API.

- Περιορισμοί ChatGPT: Παρόλο που οι απαντήσεις ήταν συνολικά αξιόπιστες, σε ερωτήσεις υψηλής πολυπλοκότητας υπήρξαν περιπτώσεις μικρών αστοχιών.

- Κόστος: Η χρήση εμπορικών API συνεπάγεται κόστος ανάλογα με τη χρήση, γεγονός που περιορίζει τη μαζική εφαρμογή.
- Πολύγλωσσο μοντέλο: Καθώς χρησιμοποιούνται υπηρεσίες streaming το API της Deepgram δεν επιτρέπει την τεχνολογία Language Detection, ώστε να γίνεται αυτόματα εναλλαγή της γλώσσας. Αυτή η τεχνολογία προς το παρόν είναι διαθέσιμη για ηχογραφημένα αρχεία και όχι για ζωντανή δειγματοληψία.

4.3 Σύγκριση με άλλες Τεχνολογίες

Για να αξιολογηθεί το σύστημα, πραγματοποιήθηκε σύγκριση με άλλες διαθέσιμες τεχνολογίες:

Πλεονεκτήματα

- Ταχύτητα απόκρισης: Το σύστημα είναι ταχύτερο από άλλες διαθέσιμες λύσεις λόγω της βελτιστοποιημένης ενσωμάτωσης API.
- Ποιότητα φωνής: Το ElevenLabs παρέχει υψηλότερη ποιότητα φωνής από άλλες εμπορικές λύσεις, με ρεαλιστική προσωδία και φυσικότητα.
- Ακρίβεια ChatGPT: Η ικανότητα κατανόησης φυσικής γλώσσας και η παραγωγή συνεπών απαντήσεων παρέχει πλεονέκτημα σε σχέση με άλλες τεχνολογίες NLP.

Μειονεκτήματα

- Κόστος: Η χρήση εμπορικών API συνεπάγεται με ανάλογο κόστος.
- Περιορισμοί σε offline χρήση: Οι λειτουργίες δεν είναι διαθέσιμες εκτός σύνδεσης, εκτός από το ChatGPT, το οποίο με τον ανάλογο προγραμματισμό και την εγκατάσταση των απαραίτητων βιβλιοθηκών μπορεί να λειτουργεί τοπικά.

Το Κεφάλαιο 4 ολοκληρώνει την αξιολόγηση του συστήματος, καταδεικνύει τα σημεία προς βελτίωση και αναδεικνύει την αξία του συστήματος σε σχέση με άλλες τεχνολογίες, με ιδιαίτερη έμφαση στη χρήση του ChatGPT.

Κεφάλαιο 5ο: Συμπεράσματα και μελλοντική βελτίωση

5.1 Συμπεράσματα

Η έρευνα απέδειξε την αποτελεσματικότητα της ενσωμάτωσης των πλέον σύγχρονων τεχνολογιών στον τομέα της τεχνητής νοημοσύνης για τη δημιουργία ενός συνεκτικού διαδραστικού συστήματος. Συνδυάζει σε ένα σύστημα τις τεχνολογίες Speech-to-Text, Natural Language Processing και Text-to-Speech, προσφέροντας μια εμπειρία για τον χρήστη που είναι όσο το δυνατόν πιο κοντά στην πραγματικότητα της ανθρώπινης επικοινωνίας.

Η βασική πλατφόρμα για αυτό το σύστημα είναι το Raspberry Pi 4, μια προσιτή, ευέλικτη πλατφόρμα για εφαρμογές που είναι διαδραστικές και ταυτόχρονα φορητές. Επιπλέον, χρησιμοποιώντας το Deepgram για τη μετατροπή φωνής σε κείμενο, το OpenAI ChatGPT για την κατανόηση φυσικής γλώσσας και το ElevenLabs για τη δημιουργία φωνής, αναπτύχθηκε μια εφαρμογή στην επίτευξη υψηλής ακρίβειας, ταχύτητας και φυσικότητας στις απαντήσεις.

Βασικά επιτεύγματα

- Αποτελεσματική επικοινωνία σε πραγματικό χρόνο:
    - Οι χρόνοι απόκρισης του συστήματος ήταν ανταγωνιστικοί, με μέσο όρο 3 έως 4 δευτερόλεπτα για την ολοκλήρωση μιας ολόκληρης αλληλεπίδρασης. Οι τεχνολογίες που χρησιμοποιήθηκαν συνέβαλαν στη μείωση των καθυστερήσεων, ενισχύοντας έτσι την ομαλή λειτουργία.
- Υψηλή ακρίβεια και ρεαλισμός:
    - Η ακρίβεια αναγνώρισης φωνής έφτασε το 98% σε περιβάλλοντα χαμηλού θορύβου, ενώ η φωνητική έξοδος χαρακτηρίστηκε από ρεαλισμό και ελκυστική προσωπικότητα.
    - Το ChatGPT παρείχε συνεπείς απαντήσεις, με ακρίβεια 85% έως 90% ακόμη και για πιο σύνθετες ερωτήσεις, καθιστώντας έτσι τη συνομιλία φυσική.
- Ευελιξία και δυνατότητα εξατομίκευσης:
    - Το σύστημα προσαρμόζεται εύκολα σε διαφορετικά περιβάλλοντα εφαρμογών, από την εκπαίδευση έως την εξυπηρέτηση πελατών. Η δυνατότητα ενσωμάτωσης περισσότερων γλωσσών και η χρήση προσαρμόσιμων API διασφαλίζουν υψηλή ευελιξία.

Προκλήσεις και περιορισμοί

Ακόμη και αν χρησιμοποιήθηκαν εκσυγχρονισμένα τεχνολογικά μέσα, το σύστημα παρουσίασε ορισμένες προκλήσεις:

- Εξάρτηση από τη σύνδεση στο Διαδίκτυο:
    - Η λειτουργία του εξαρτάται από τη συνεχή σύνδεση με τα API, περιορίζοντας έτσι τη χρήση του σε περιβάλλοντα χωρίς διαθέσιμο δίκτυο.
- Περιορισμοί πολυγλωσσίας:
    - Παρόλο που οι υπηρεσίες παρέχονται σε πολλές γλώσσες, δεν υπάρχει αυτόματη ανίχνευση και εναλλαγή γλωσσών κατά τη διάρκεια ζωντανών μεταδόσεων, ορίζοντας έτσι την γλώσσα κατά τον προγραμματισμό της υλοποίησης.
- Κόστος χρήσης:
    - Η χρήση εμπορικών APIs αυξάνει το συνολικό κόστος λειτουργίας, γεγονός που μπορεί ενδεχομένως να περιορίσει την μαζική εφαρμογή του συστήματος, ιδιαίτερα σε έργα με περιορισμένο προϋπολογισμό.

Η έρευνα αυτή συνέβαλε σημαντικά στον τομέα της τεχνητής νοημοσύνης και των διαδραστικών συστημάτων με την εισαγωγή ενός πλήρως λειτουργικού πρωτοτύπου με δυνατότητες προσαρμογής και ανάπτυξης. Η πρωτοποριακή εφαρμογή και η ενσωμάτωση όλων αυτών των τεχνολογιών έδειξε ότι είναι εφικτή η κατασκευή ενός πολυδιάστατου συστήματος που θα μπορούσε να χρησιμοποιηθεί για διαφορετικές εφαρμογές με τις ιδιαίτερες ανάγκες τους, όπως:

- Εκπαίδευση: ψηφιακοί βοηθοί για παιδαγωγικούς σκοπούς
- Υγεία: Υποστήριξη ατόμων με αναπηρία με τεχνολογία φωνητικών εντολών.
- Εξυπηρέτηση πελατών: Βελτιωμένη εμπειρία πελατών μέσω διαδραστικής εξυπηρέτησης πελατών.

Το σύστημα μπορεί να χρησιμεύσει ως πρότυπο για μελλοντικές υλοποιήσεις, επειδή συνδυάζει προηγμένη τεχνολογία με απλότητα στο σχεδιασμό και εξοικονόμηση πόρων.

5.2 Μελλοντικές βελτιώσεις και αναβαθμίσεις

Σημαντική κατεύθυνση για τη μελλοντική εργασία είναι η βελτίωση της πολυγλωσσικότητας του συστήματος. Ειδικότερα, απαιτείται η ανάπτυξη μεθόδων που θα επιτρέπουν την αυτόματη ανίχνευση γλώσσας σε ζωντανή μετάδοση, εξαλείφοντας τις αστάθειες και επιτρέποντας τη δυναμική εναλλαγή μεταξύ διαφορετικών γλωσσών. Αυτή η προσέγγιση θα ενισχύσει τη φιλικότητα του συστήματος προς τον χρήστη και θα διευρύνει το πεδίο εφαρμογών του σε παγκόσμια κλίμακα.

Ένας ακόμη στόχος είναι η ενσωμάτωση ρομποτικής όρασης μέσω της προσθήκης κάμερας, η οποία θα λειτουργεί ως "ρομποτικά μάτια". Με την ενσωμάτωση τεχνολογιών αναγνώρισης αντικειμένων μέσω AI, το σύστημα θα αποκτήσει τη δυνατότητα ανάλυσης του περιβάλλοντος και παροχής πληροφοριών στον χρήστη βάσει εικόνων. Η χρήση του ChatGPT ή άλλων προηγμένων τεχνολογιών για την ανάλυση των εικόνων θα ενισχύσει την ευφυΐα του συστήματος, επιτρέποντας την εκτέλεση πιο σύνθετων λειτουργιών και τη δημιουργία εμπειρίας που προσεγγίζει την ανθρώπινη αντίληψη, καθώς

πλέον θα μπορεί να λαμβάνει επιπλέον ερεθίσματα από τον περιβάλλοντα χώρο αλλά και ο χρήστης θα μπορεί να δείξει ένα αντικείμενο αντί να το περιγράψει. Επιπλέον, με την χρήση της κάμερας μπορούν να υλοποιηθούν επιπλέον λειτουργίες όπως εκτίμηση διαστάσεων και αποστάσεων αλλά και αναγνώριση νοηματικής γλώσσας κάνοντας το σύστημα συμβατό και για άτομα με αναπηρία.

Η βελτίωση της αυτονομίας του συστήματος αποτελεί επίσης προτεραιότητα. Η ενσωμάτωση λειτουργιών offline, μέσω τοπικών μοντέλων, θα μειώσει την εξάρτηση από τη σύνδεση στο διαδίκτυο και θα επιτρέψει τη χρήση του συστήματος σε απομακρυσμένες περιοχές ή περιβάλλοντα με περιορισμένη συνδεσιμότητα. Επιπλέον, η ενεργειακή αποδοτικότητα του συστήματος μπορεί να βελτιωθεί περαιτέρω, καθιστώντας το ιδανικό για χρήση σε φορητές συσκευές και εφαρμογές που απαιτούν χαμηλή κατανάλωση ενέργειας.

Η προσθήκη συναισθηματικής νοημοσύνης είναι μια πολλά υποσχόμενη κατεύθυνση. Η ανίχνευση συναισθημάτων μέσω φωνής ή εκφράσεων θα επιτρέψει τη δημιουργία πιο φυσικών και ανθρώπινων αλληλεπιδράσεων. Τέλος, η εισαγωγή παραμετροποιήσιμων χαρακτηριστικών θα προσαρμόσει το σύστημα στις εξατομικευμένες ανάγκες του κάθε χρήστη, καθιστώντας το ακόμα πιο ευέλικτο και χρηστικό.

Ενσωμάτωση επιπλέον APIs για περισσότερες λειτουργίες και παροχές, όπως καιρικές προβλέψεις βάσει της υπάρχουσας τοποθεσίας , αναπαραγωγή μουσικής ενσωματώνοντας το API του Spotify αλλά και την χρήση των μέσων κοινωνικής δικτύωσης μέσω φωνητικών εντολών.

Με την υλοποίηση αυτών των βελτιώσεων, το σύστημα θα εξελιχθεί σε μια ολοκληρωμένη πλατφόρμα αιχμής, ικανή να ανταποκριθεί στις προκλήσεις του μέλλοντος και να επαναπροσδιορίσει τη χρήση των διαλογικών συστημάτων στην καθημερινότητα όλων τον ατόμων ακόμη και αυτών με κάποια μορφή αναπηρίας.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] W. Liu, G. Zhuang, X. Liu, S. Hu, R. He και Y. and Wang, «How do we move towards true artificial intelligence,» *2021 IEEE 23rd International Conference on High Performance Computing & Communications; 7th International Conference on Data Science & Systems; 19th International Conference on Smart City; 7th International Conference on Dependability in Sensor, Cloud & B,* 20-22 December 2021.

[2] Z. Han, «The Application of Artificial Intelligence in Computer Network Technology,» *2021 2nd International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT),* pp. 632-635, 2021.

[3] K.-L. A. YAU, H. J. LEE, Y.-W. CHONG, M. H. LING, A. R. SYED, C. WU και H. G. and GOH, «Augmented Intelligence: Surveys of Literature and Expert Opinion to Understand Relations Between Human Intelligence and Artificial Intelligence,» *IEEE Access,* pp. 136744-136761, 24 September 2021.

[4] J. Harika, P. Baleeshwar, K. Navya και a. H. Shanmugasundaram, «A Review on Artificial Intelligence with Deep Human Reasoning,» *Proceedings of the International Conference on Applied Artificial Intelligence and Computing (ICAAIC 2022),* pp. 81-84, 2022.

[5] O. K. a. I. G. Harris, «Chatbot-based assertion generation from natural language specifications,» *2019 Forum for Specification and Design Languages (FDL),* pp. 1-6, 2019.

[6] K. J. a. X. Lu, «Natural Language Processing and Its Applications in Machine Translation: A Diachronic Review,» *2020 IEEE 3rd International Conference on Safe Production and Informatization (IICSPI),* pp. 210-214, 2020.

[7] V. Hristidis, «Chatbot Technologies and Challenges,» *2018 First International Conference on Artificial Intelligence for Industries (AI4I),* pp. 126-126, 2018.

[8] S. K. Maher, S. G. Bhable, A. R. Lahase και a. S. S. Nimbhore, «AI and Deep Learning-driven Chatbots: A Comprehensive Analysis and Application Trends,» *2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS),* pp. 994-998, 2022.

[9] A. Abdellatif, K. Badran, D. E. Costa και a. E. Shihab, «A Comparison of Natural Language Understanding Platforms for Chatbots in Software Engineering,» *IEEE Transactions on Software Engineering,* τόμ. 48, αρ. 8, pp. 3087-3102, 1 8 2022.

[10] H. Abdulla, A. M. Eltahir, S. Alwahaishi, K. Saghair, J. Platos και a. V. Snasel, «Chatbots Development Using Natural Language Processing: A Review,» *2022 26th International Conference on Circuits, Systems, Communications and Computers (CSCC),* pp. 122-128, 2022.

[11] OpenAI, «ChatGPT,» OpenAI, 2025. [Ηλεκτρονικό]. Available: https://openai.com/chatgpt.

[12] DeepMind, «Gemini,» Google,DeepMind, 2025. [Ηλεκτρονικό]. Available: https://deepmind.google/technologies/gemini.

[13] Anthropic, «Anthropic,» Anthropic, 2025. [Ηλεκτρονικό]. Available: https://www.anthropic.com.

[14] V. M. Reddy, T. Vaishnavi και a. K. P. Kumar, «Speech-to-Text and Text-to-Speech Recognition Using Deep Learning,» *2023 2nd International Conference on Edge Computing and Applications (ICECAA),* pp. 657-666, 2023.

[15] Y. H. Ghadage και a. S. D. Shelke, «Speech to text conversion for multilingual languages,» *2016 International Conference on Communication and Signal Processing (ICCSP),* pp. 0236-0240, 2016.

[16] Deepgram, «Deepgram Documentation - Model,» Deepgram, 2025. [Ηλεκτρονικό]. Available: https://developers.deepgram.com/docs/model.

[17] ElevenLabs, «ElevenLabs Documentation - Text-to-Speech,» ElevenLabs, 2025. [Ηλεκτρονικό]. Available: https://elevenlabs.io/docs/product/introduction#text-to-speech.

[18] OpenAI, «About OpenAI,» OpenAI, 2025. [Ηλεκτρονικό]. Available: https://openai.com/about.

[19] A. Radford, K. Narasimhan, T. Salimans και a. I. Sutskever, «Improving Language Understanding by Generative Pre-Training,» OpenAI.

[20] OpenAI, «Whisper,» OpenAI, 2025. [Ηλεκτρονικό]. Available: https://openai.com/index/whisper/.

[21] OpenAI, «DALL·E,» OpenAI, 2025. [Ηλεκτρονικό]. Available: https://openai.com/index/dall-e/.

[22] OpenAI, «OpenAI Codex,» OpenAI, 2025. [Ηλεκτρονικό]. Available: https://openai.com/index/openai-codex/.

[23] GitHub, «GitHub Copilot,» GitHub, 2025. [Ηλεκτρονικό]. Available: https://github.com/features/copilot.

[24] OpenAI, «OpenAI Platform Documentation - Overview,» OpenAI, 2025. [Ηλεκτρονικό]. Available: https://platform.openai.com/docs/overview.

[25] OpenAI, «Spinning Up in Deep Reinforcement Learning,» OpenAI, 2025. [Ηλεκτρονικό]. Available: https://spinningup.openai.com/en/latest/.

[26] N. Zhang, Y. Zou, X. Xia, Q. Huang, D. Lo και a. S. Li, «Web APIs: Features, Issues, and Expectations – A Large-Scale Empirical Study of Web APIs From Two Publicly Accessible Registries Using Stack Overflow and a User Survey,» *IEEE Transactions on Software Engineering*, τόμ. 49, αρ. 2, pp. 498-528, 1 2 2023.

[27] S. Jonnada και a. J. K. Joy, «Measure your API Complexity and Reliability,» *2019 IEEE 17th International Conference on Software Engineering Research, Management and Applications (SERA)*, pp. 104-109, 2019.

[28] A. H. e. al., «The Long Tail: Understanding the Discoverability of API Functionality,» *2019 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pp. 157-161, 2019.

[29] Isha, A. Sharma και a. M. Revathi, «Automated API Testing,» *2018 3rd International Conference on Inventive Computation Technologies (ICICT)*, pp. 788-791, 2018.

[30] R. Pi, «Raspberry Pi 4 Model B,» Raspberry Pi Foundation, 2025. [Ηλεκτρονικό]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/.

[31] Vention, «USB External Sound Card,» Vention Technology, 2025. [Ηλεκτρονικό]. Available: https://ventiontech.com/products/usb-external-sound-card.

[32] HyperX, «HyperX Cloud II Gaming Headset Microphone Specification,» HyperX, 2025. [Ηλεκτρονικό]. Available: https://hyperx.com/products/hyperx-cloud-ii?variant=40998813827229.

[33] SanDisk, «SanDisk Ultra UHS-I microSD Card,» SanDisk, 2025. [Ηλεκτρονικό]. Available: https://shop.sandisk.com/el-gr/products/memory-cards/microsd-cards/sandisk-ultra-uhs-i-microsd?sku=SDSQUA4-032G-GN6MA.

[34] Philips, «Philips SPA3210/10 Multimedia Speakers 2.0,» Philips, 2025. [Ηλεκτρονικό]. Available: https://www.philips.gr/c-p/SPA3210_10/multimedia-speakers-2.0.

[35] V. Cutting και a. N. Stephen, «A Review on using Python as a Preferred Programming Language for Beginners,» *International Research Journal of Engineering and Technology (IRJET)*, τόμ. 08, αρ. 08, pp. 4258-4263, 8 2021.

ΠΑΡΑΡΤΗΜΑ Α : Πλήρης Κώδικας

```
from Deepgram import Speech_to_Text

from OpenAI import ask_chatgpt, conversation_history
import speech_recognition as sr
from dotenv import load_dotenv
import os
import openai
from deepgram import DeepgramClient
from elevenlabs.client import ElevenLabs
from elevenlabs import stream , VoiceSettings
import time

load_dotenv()

DEEPGRAM_API_KEY = os.getenv('deepgram_key')
OPENAI_API_KEY = os.getenv('api_data')
ELEVENLABS_API_KEY = os.getenv('eleven_data')

deepgram: DeepgramClient = DeepgramClient(DEEPGRAM_API_KEY)
openai.api_key = OPENAI_API_KEY
client = ElevenLabs(api_key= ELEVENLABS_API_KEY)

def text_to_speech(input):
    audio_stream = client.generate(
    text=input,
    stream=True,
    model='eleven_multilingual_v2',
    voice="TX3LPaxmHKxFdv7VOQHJ",
    )
    stream(audio_stream)


def main():
    recognizer = sr.Recognizer()
    microphone = sr.Microphone()
```

```
    text_to_speech("Welcome to Jarvis project....Please wait!")
    time.sleep(3)
    text_to_speech("You can say 'Hey jarvis!' or 'Hello Jarvis!' and i'll be here for
you!")
    while True:
        try:
            with microphone as source:
                recognizer.adjust_for_ambient_noise(source, duration=0.4)
                print("Listening... Please speak.")
                audio = recognizer.listen(source, timeout=5, phrase_time_limit=5)  #
προσθήκη timeouts
                text = recognizer.recognize_google(audio)
                print(f"You said: {text}")
                if "hey jarvis" in text.lower() or "hello jarvis" in text.lower():
                    text_to_speech("Hello sir! how can i help you today?")
                    while True:
                        question = Speech_to_Text(deepgram)
                        answer = ask_chatgpt(question)
                        text_to_speech(answer)
                        if "goodbye" in question.lower() or "thank you" in
question.lower():
                          conversation_history = [
                                {
                                    "role": "system",
                                    "content": "Your name is Jarvis. You are a really kind,
helpful, funny, witty, full of personality and trustful assistant who helps people solve
their problems and answer to their questions. Your creator is Thanasis Palaiogiannis and he
made you for his bachelor's Thesis. Whenever he tells you goodbye or thank you, it means
that the conversation stops, so you greet and close. The answers might be relatively short
and comprehensive.In this project you are not a text conversation ai but you are a humanoid
speaker so you can hear what you actually read. "
                                }
                            ]  # Καθαρίζουμε τη μνήμη της συνομιλίας μόνο στο τέλος
                          break
                elif "shut down" in text.lower():
                    break

        except sr.UnknownValueError:
            print("I could not understand you. Please speak again.")
        except sr.RequestError as e:
            print(f"Could not request results from Google Speech Recognition service; {e}")
        except Exception as e:
            print(f"An unexpected error occurred: {e}")

if __name__ == "__main__":
    main()
#trigger phrase code
import speech_recognition as sr
from ElevenLabss import tts
```

```
def activation_phrase():
    recognizer = sr.Recognizer()
    microphone = sr.Microphone()

    while True:
        try:
            with microphone as source:
                recognizer.adjust_for_ambient_noise(source, duration=0.4)
                print("Listening... Please speak.")
                audio = recognizer.listen(source, timeout=5, phrase_time_limit=5)  #
προσθήκη timeouts
                text = recognizer.recognize_google(audio)
                print(f"You said: {text}")
                if "hey jarvis" in text.lower() or "hello jarvis" in text.lower():
                    tts("Hello sir! how can i help you today?",)
                elif "goodbye" in text.lower():
                    tts("Goodbye sir!")
                    break
        except sr.UnknownValueError:
            print("I could not understand you. Please speak again.")
        except sr.RequestError as e:
            tts(f"Could not request results from Google Speech Recognition service; {e}")
        except Exception as e:
            print(f"An unexpected error occurred: {e}")


#Deepgramm Code
import threading
from deepgram import (
    LiveTranscriptionEvents,
    LiveOptions,
    Microphone,
)

def Speech_to_Text(deepgram):
    is_finals = []
    final_text = ""
    finish_event = threading.Event()  # Χρησιμοποιούμε ένα Event για ασφαλή τερματισμό

    try:
        dg_connection = deepgram.listen.websocket.v("1")

        def on_open(self, open, **kwargs):
            print("Connection Open")

        def on_message(self, result, **kwargs):
            nonlocal is_finals
```

```
        nonlocal final_text
        sentence = result.channel.alternatives[0].transcript
        if len(sentence) == 0:
            return
        if result.is_final:
            print(f"Message: {result.to_json()}")
            is_finals.append(sentence)

            if result.speech_final:
                utterance = " ".join(is_finals)
                final_text = utterance
                print(f"Speech Final: {utterance}")
                is_finals = []
                # Set event to indicate we're done
                finish_event.set()
        else:
            print(f"Interim Results: {sentence}")

    def on_metadata(self, metadata, **kwargs):
        print(f"Metadata: {metadata}")

    def on_speech_started(self, speech_started, **kwargs):
        print("Speech Started")

    def on_utterance_end(self, utterance_end, **kwargs):
        nonlocal is_finals
        if len(is_finals) > 0:
            utterance = " ".join(is_finals)
            print(f"Utterance End: {utterance}")
            is_finals = []

    def on_close(self, close, **kwargs):
        print("Connection Closed")

    def on_error(self, error, **kwargs):
        print(f"Handled Error: {error}")

    def on_unhandled(self, unhandled, **kwargs):
        print(f"Unhandled Websocket Message: {unhandled}")

    dg_connection.on(LiveTranscriptionEvents.Open, on_open)
    dg_connection.on(LiveTranscriptionEvents.Transcript, on_message)
    dg_connection.on(LiveTranscriptionEvents.Metadata, on_metadata)
    dg_connection.on(LiveTranscriptionEvents.SpeechStarted, on_speech_started)
    dg_connection.on(LiveTranscriptionEvents.UtteranceEnd, on_utterance_end)
    dg_connection.on(LiveTranscriptionEvents.Close, on_close)
    dg_connection.on(LiveTranscriptionEvents.Error, on_error)
    dg_connection.on(LiveTranscriptionEvents.Unhandled, on_unhandled)
```

```
        options: LiveOptions = LiveOptions(
            model="nova-2",
            language="en",
            smart_format=True,
            encoding="linear16",
            channels=1,
            sample_rate=16000,
            interim_results=True,
            utterance_end_ms="1000",
            vad_events=True,
            endpointing=300,
        )

        addons = {
            "no_delay": "true"
        }

        if dg_connection.start(options, addons=addons) is False:
            print("Failed to connect to Deepgram")
            return None

        microphone = Microphone(dg_connection.send)
        microphone.start()

        # Χρησιμοποιούμε το Event για να περιμένουμε τον τερματισμό με ασφάλεια
        finish_event.wait()  # Περιμένει μέχρι να οριστεί το event από το τελικό κείμενο

        # Όταν ολοκληρωθεί, κλείνουμε τα πάντα με ασφάλεια
        microphone.finish()
        dg_connection.finish()

        return final_text
    except Exception as e:
        print(f"Could not open socket: {e}")
        return None

if __name__ == "__main__":
    text = Speech_to_Text()
    if text:
        print(f"Final Recognized Text: {text}")
    else:
        print("No text was recognized or an error occurred.")
```