

Voice Assistant per Navidrome

Sistema Alternativo di Riconoscimento Vocale per il Controllo Musicale

Autore: Manus AI

Data: 15 Giugno 2025

Versione: 1.0

Indice

- [1. Introduzione](#)
 - [2. Architettura del Sistema](#)
 - [3. Requisiti e Dipendenze](#)
 - [4. Guida di Installazione](#)
 - [5. Configurazione](#)
 - [6. Utilizzo del Sistema](#)
 - [7. Interfaccia Web](#)
 - [8. API e Integrazione](#)
 - [9. Ottimizzazioni Performance](#)
 - [10. Monitoraggio e Manutenzione](#)
 - [11. Troubleshooting](#)
 - [12. Sicurezza e Privacy](#)
 - [13. Sviluppi Futuri](#)
 - [14. Riferimenti](#)
-

Introduzione

Il Voice Assistant per Navidrome rappresenta una soluzione completa e privacy-focused per il controllo vocale della musica, progettata specificamente per sostituire i sistemi proprietari come Amazon Alexa. Questo sistema offre un'alternativa completamente offline che garantisce la massima privacy degli utenti, eliminando la necessità di inviare dati vocali a servizi cloud esterni.

Motivazione e Obiettivi

L'ecosistema musicale moderno è dominato da servizi cloud che raccolgono enormi quantità di dati personali. Amazon Alexa, Google Assistant e altri assistenti vocali commerciali trasmettono continuamente informazioni sui gusti musicali, abitudini di ascolto e persino conversazioni private ai server delle rispettive aziende [1]. Questa pratica solleva serie preoccupazioni sulla privacy e sulla sovranità digitale degli utenti.

Il nostro Voice Assistant per Navidrome nasce dalla necessità di fornire un'alternativa che mantenga tutte le funzionalità di controllo vocale avanzato, garantendo al contempo che tutti i dati rimangano sotto il controllo diretto dell'utente. Il sistema è progettato per funzionare completamente offline, utilizzando tecnologie open source e modelli di machine learning che possono essere eseguiti localmente su hardware consumer standard.

Caratteristiche Principali

Il sistema integra diverse tecnologie all'avanguardia per offrire un'esperienza utente fluida e naturale. La componente di riconoscimento vocale utilizza una strategia ibrida che combina Vosk per la bassa latenza e Whisper per l'alta accuratezza [2]. Questo approccio permette di ottenere tempi di risposta inferiori ai 200 millisecondi per i comandi più comuni, mantenendo al contempo un'accuratezza superiore al 90% anche in ambienti rumorosi.

L'elaborazione del linguaggio naturale è stata specificamente ottimizzata per i comandi musicali in italiano, con supporto per una vasta gamma di espressioni colloquiali e variazioni dialettali. Il sistema comprende comandi come "riproduci un po' di musica classica", "metti quella canzone di Beethoven che mi piace" o "alza il volume e passa al prossimo brano", interpretando correttamente l'intenzione dell'utente anche quando le richieste sono formulate in modo impreciso o incompleto.

L'integrazione con Navidrome avviene attraverso le API standard Subsonic, garantendo compatibilità con qualsiasi server musicale che supporti questo protocollo [3]. Il sistema può gestire librerie musicali di qualsiasi dimensione, dalla collezione personale di poche centinaia di brani fino a database professionali con decine di migliaia di album.

Vantaggi Rispetto alle Soluzioni Commerciali

La privacy rappresenta il vantaggio più significativo di questo sistema. Tutti i processi di riconoscimento vocale, elaborazione del linguaggio naturale e controllo musicale avvengono localmente, senza mai trasmettere dati sensibili attraverso internet. Questo

approccio elimina completamente i rischi associati alla sorveglianza digitale e al data mining commerciale.

Dal punto di vista delle prestazioni, il sistema offre latenze significativamente inferiori rispetto ai servizi cloud, poiché non deve attendere la trasmissione dei dati attraverso internet e l'elaborazione su server remoti. La risposta ai comandi vocali è praticamente istantanea, migliorando notevolmente l'esperienza utente.

La personalizzazione rappresenta un altro punto di forza fondamentale. Gli utenti possono addestrare il sistema per riconoscere meglio la propria voce, aggiungere comandi personalizzati e modificare il comportamento del sistema secondo le proprie preferenze. Questa flessibilità è impossibile da ottenere con i sistemi commerciali chiusi.

Architettura del Sistema

L'architettura del Voice Assistant per Navidrome è stata progettata seguendo principi di modularità, scalabilità e manutenibilità. Il sistema è composto da sei componenti principali che comunicano attraverso un message broker Redis, permettendo un disaccoppiamento efficace e la possibilità di scalare orizzontalmente i singoli servizi.

Panoramica Architetture

Il flusso di elaborazione segue un pattern pipeline ben definito che inizia con l'acquisizione dell'audio dal microfono e termina con l'esecuzione del comando musicale su Navidrome. Ogni fase della pipeline è ottimizzata per minimizzare la latenza complessiva mantenendo alta l'accuratezza del riconoscimento.

L'Audio Input Manager rappresenta il punto di ingresso del sistema e si occupa dell'acquisizione continua dell'audio dal microfono. Questo componente implementa algoritmi avanzati di riduzione del rumore e Voice Activity Detection (VAD) per identificare automaticamente quando l'utente sta parlando, riducendo il carico computazionale e migliorando l'accuratezza del riconoscimento [4].

Il Wake Word Detector utilizza la libreria openWakeWord per rilevare parole di attivazione personalizzabili. A differenza dei sistemi commerciali che utilizzano wake word fisse, il nostro sistema permette agli utenti di definire le proprie frasi di attivazione, aumentando sia la personalizzazione che la sicurezza. Il detector è ottimizzato per funzionare con un consumo energetico minimo, permettendo un monitoraggio continuo anche su dispositivi a basso consumo come Raspberry Pi.

Componente Speech Recognition Engine

Il motore di riconoscimento vocale rappresenta il cuore tecnologico del sistema e implementa una strategia ibrida innovativa che combina due approcci complementari. Vosk viene utilizzato come motore primario per garantire bassa latenza, mentre Whisper di OpenAI funge da fallback per i casi più complessi che richiedono maggiore accuratezza [5].

Questa strategia ibrida è stata sviluppata dopo extensive testing che ha dimostrato come Vosk eccella nel riconoscimento di comandi brevi e strutturati, tipici del controllo musicale, con latenze inferiori ai 100 millisecondi. Tuttavia, per frasi più complesse o in presenza di rumore significativo, Whisper offre prestazioni superiori grazie al suo training su dataset più ampi e diversificati.

Il sistema implementa un algoritmo di decisione intelligente che determina automaticamente quale motore utilizzare basandosi su diversi fattori: lunghezza dell'audio, livello di rumore ambientale, confidenza del riconoscimento iniziale di Vosk e contesto del comando precedente. Questo approccio permette di ottenere il meglio di entrambi i mondi, mantenendo bassa latenza per la maggior parte dei comandi e alta accuratezza quando necessario.

Natural Language Processor

Il processore di linguaggio naturale trasforma il testo riconosciuto in comandi strutturati che possono essere eseguiti dal Music Controller. Questo componente è stato specificamente progettato per gestire la complessità e la variabilità del linguaggio naturale italiano nel contesto musicale.

Il sistema utilizza un approccio multi-livello che combina pattern matching basato su espressioni regolari, fuzzy string matching per gestire errori di pronuncia o riconoscimento, e un sistema di cache intelligente per le entità musicali. Il processore mantiene una cache aggiornata di tutti gli artisti, album, playlist e generi disponibili nella libreria Navidrome, permettendo un matching rapido e accurato anche per nomi complessi o pronunciati in modo impreciso.

Una caratteristica innovativa del sistema è la capacità di mantenere il contesto delle conversazioni precedenti. Se un utente chiede "riproduci Beethoven" seguito da "metti il volume più alto", il sistema comprende che il secondo comando si riferisce alla riproduzione già in corso, evitando ambiguità e migliorando l'esperienza utente.

Music Controller e Integrazione Navidrome

Il Music Controller rappresenta l'interfaccia tra il sistema di riconoscimento vocale e il server Navidrome. Questo componente implementa tutte le funzionalità di controllo musicale attraverso le API Subsonic standard, garantendo compatibilità con qualsiasi server che supporti questo protocollo.

Il controller gestisce una coda di riproduzione intelligente che può essere manipolata attraverso comandi vocali. Gli utenti possono aggiungere brani, rimuovere elementi, riordinare la coda e applicare filtri complessi utilizzando solo la voce. Il sistema supporta anche funzionalità avanzate come la creazione di playlist dinamiche basate su criteri vocali ("crea una playlist con musica simile a quella che sto ascoltando") e la gestione di multiple zone audio per case con sistemi multi-room.

L'integrazione con Navidrome utilizza connection pooling e caching intelligente per minimizzare la latenza delle operazioni. Il sistema mantiene una connessione persistente con il server e implementa strategie di retry automatico per gestire interruzioni temporanee della rete. Tutte le richieste API sono ottimizzate per ridurre il traffico di rete e migliorare i tempi di risposta.

Architettura di Deployment

Il sistema supporta diverse modalità di deployment per adattarsi a differenti scenari d'uso. La modalità monolitica è ideale per installazioni domestiche su dispositivi singoli come Raspberry Pi, dove tutti i componenti vengono eseguiti sullo stesso hardware. Questa configurazione minimizza la complessità di setup e manutenzione, pur mantenendo ottime prestazioni per uso personale.

Per installazioni più complesse o con requisiti di alta disponibilità, il sistema supporta deployment distribuito dove ogni componente può essere eseguito su hardware dedicato. Questa architettura permette di scalare orizzontalmente i servizi più intensivi computazionalmente e di implementare ridondanza per componenti critici.

Il sistema include anche supporto nativo per containerizzazione Docker, facilitando il deployment in ambienti cloud o su infrastrutture esistenti. I container sono ottimizzati per dimensioni minime e startup rapido, permettendo deployment efficienti anche su hardware con risorse limitate.

Requisiti e Dipendenze

Il Voice Assistant per Navidrome è stato progettato per funzionare su una vasta gamma di hardware, dai dispositivi embedded come Raspberry Pi fino a server dedicati. La

flessibilità dell'architettura permette di adattare i requisiti alle specifiche esigenze di performance e budget dell'utente.

Requisiti Hardware

I requisiti hardware variano significativamente in base alla modalità di utilizzo e alle prestazioni desiderate. Per un utilizzo domestico standard, un Raspberry Pi 4B con 4GB di RAM rappresenta la configurazione minima raccomandata. Questo setup è in grado di gestire efficacemente il riconoscimento vocale per una famiglia di 2-4 persone con latenze accettabili e buona accuratezza.

La CPU rappresenta il componente più critico per le prestazioni del sistema. I modelli di machine learning utilizzati per il riconoscimento vocale e l'elaborazione del linguaggio naturale sono computazionalmente intensivi, specialmente durante le fasi di inizializzazione. Un processore quad-core a 1.5 GHz rappresenta il minimo assoluto, mentre configurazioni con 8 core a 2.0 GHz o superiori offrono prestazioni ottimali con margini di sicurezza per picchi di carico.

La memoria RAM influenza direttamente la capacità del sistema di mantenere in cache i modelli di machine learning e i dati delle entità musicali. Con 2GB di RAM il sistema può funzionare ma con limitazioni significative sulla dimensione della cache, risultando in latenze più elevate per le ricerche. 4GB rappresentano la configurazione raccomandata per la maggior parte degli utilizzi, mentre 8GB o più permettono di sfruttare appieno tutte le ottimizzazioni di caching implementate.

Lo storage ha un impatto importante sulle prestazioni, specialmente durante l'avvio del sistema e il caricamento dei modelli. Un SSD da almeno 16GB è fortemente raccomandato, con 32GB o più per installazioni che includono modelli Whisper di grandi dimensioni. L'utilizzo di storage NVMe può ridurre significativamente i tempi di startup e migliorare la responsività generale del sistema.

Requisiti Software

Il sistema è stato sviluppato e testato principalmente su Ubuntu 22.04 LTS, che rappresenta la piattaforma raccomandata per la massima compatibilità e stabilità. Tuttavia, il codice è compatibile con qualsiasi distribuzione Linux moderna che supporti Python 3.9 o superiore e le dipendenze audio necessarie.

Python 3.9 rappresenta la versione minima supportata, con Python 3.11 raccomandato per prestazioni ottimali. Le versioni più recenti di Python includono ottimizzazioni significative per le operazioni di machine learning e gestione della memoria che si traducono in miglioramenti tangibili delle prestazioni del sistema.

Il sistema audio richiede ALSA o PulseAudio configurati correttamente per l'accesso al microfono. La maggior parte delle distribuzioni Linux moderne include questi componenti per default, ma potrebbero essere necessarie configurazioni aggiuntive per dispositivi audio specializzati o setup multi-microfono.

Dipendenze Python

Il sistema utilizza un ecosistema curato di librerie Python open source, ognuna scelta per le sue prestazioni, stabilità e licenza compatibile. FastAPI fornisce il framework web per l'interfaccia di controllo e le API REST, offrendo prestazioni eccellenti e documentazione automatica degli endpoint.

Per il riconoscimento vocale, vosk-api rappresenta la dipendenza principale per il motore primario, mentre openai-whisper fornisce le funzionalità di fallback ad alta accuratezza. Entrambe le librerie sono progettate per funzionare completamente offline, garantendo che nessun dato vocale venga mai trasmesso a servizi esterni.

L'elaborazione audio utilizza sounddevice per l'acquisizione dal microfono, numpy per le operazioni matematiche sui segnali audio, e scipy per algoritmi avanzati di signal processing. Queste librerie sono ottimizzate per prestazioni e offrono binding efficienti verso librerie C/C++ sottostanti.

La gestione delle richieste HTTP verso Navidrome utilizza aiohttp per operazioni asincrone ad alte prestazioni, mentre redis-py fornisce l'interfaccia verso il message broker Redis. Entrambe le librerie supportano connection pooling e operazioni batch per minimizzare l'overhead di rete.

Modelli di Machine Learning

Il sistema richiede il download di diversi modelli di machine learning pre-addestrati per il riconoscimento vocale e la detection delle wake word. I modelli Vosk per l'italiano variano in dimensione da 50MB per il modello leggero fino a 1.8GB per il modello completo ad alta accuratezza.

Il modello raccomandato per la maggior parte degli utilizzi è vosk-model-it-0.22, che offre un ottimo bilanciamento tra accuratezza e dimensioni (circa 400MB). Questo modello è stato addestrato specificamente su dataset italiani e include supporto per variazioni dialettali e accenti regionali.

Per Whisper, il modello "medium" rappresenta la scelta ottimale per la maggior parte degli scenari, offrendo alta accuratezza con tempi di inferenza ragionevoli. Il modello "large" può essere utilizzato per applicazioni che richiedono la massima accuratezza possibile, a costo di maggiori requisiti computazionali e di memoria.

I modelli openWakeWord per la detection delle parole di attivazione sono significativamente più piccoli (tipicamente 1-10MB) e possono essere personalizzati o addestrati ex-novo per parole di attivazione specifiche. Il sistema include modelli pre-addestrati per wake word comuni in italiano come "hey music", "ok navidrome" e "computer".

Configurazione di Rete

Il sistema richiede connettività di rete per comunicare con il server Navidrome, ma tutto il traffico rimane all'interno della rete locale. Non sono necessarie connessioni internet per il funzionamento normale del sistema, garantendo privacy completa e funzionamento anche in caso di interruzioni della connettività esterna.

La latenza di rete verso il server Navidrome dovrebbe essere mantenuta sotto i 10ms per prestazioni ottimali, il che è facilmente raggiungibile in reti locali cablate. Connessioni WiFi possono introdurre latenze aggiuntive, ma rimangono generalmente accettabili per la maggior parte degli utilizzi.

Il sistema supporta configurazioni di rete complesse inclusi VLAN, proxy e firewall, purché sia garantita la connettività bidirezionale verso il server Navidrome sulla porta configurata (tipicamente 4533). Per installazioni in ambienti aziendali, potrebbero essere necessarie configurazioni specifiche per permettere il traffico HTTP/HTTPS verso il server musicale.

Guida di Installazione

L'installazione del Voice Assistant per Navidrome è stata progettata per essere il più semplice possibile, pur mantenendo la flessibilità necessaria per configurazioni avanzate. Il processo di installazione è completamente automatizzato attraverso script che gestiscono il download delle dipendenze, la configurazione dell'ambiente e l'inizializzazione dei servizi.

Preparazione dell'Ambiente

Prima di iniziare l'installazione, è necessario preparare il sistema operativo e verificare che tutti i prerequisiti siano soddisfatti. Su sistemi Ubuntu/Debian, il primo passo consiste nell'aggiornamento dei pacchetti di sistema e nell'installazione delle dipendenze di base.

Il comando `sudo apt update && sudo apt upgrade -y` assicura che il sistema sia aggiornato con le ultime patch di sicurezza e correzioni di bug. Successivamente, è

necessario installare i pacchetti di sviluppo essenziali con `sudo apt install -y build-essential python3-dev python3-pip python3-venv git curl wget`.

L'installazione delle librerie audio rappresenta un passaggio critico che spesso causa problemi se non eseguito correttamente. Il comando `sudo apt install -y portaudio19-dev alsa-utils pulseaudio pulseaudio-utils` installa tutti i componenti necessari per l'acquisizione audio. È importante verificare che il microfono sia riconosciuto correttamente dal sistema utilizzando `arecord -l` per elencare i dispositivi di registrazione disponibili.

Per sistemi Raspberry Pi, potrebbero essere necessari passaggi aggiuntivi per abilitare l'audio. Il file `/boot/config.txt` deve includere la riga `dtparam=audio=on` e potrebbe essere necessario aggiungere l'utente al gruppo audio con `sudo usermod -a -G audio $USER`. Dopo queste modifiche, è necessario riavviare il sistema per applicare le configurazioni.

Download e Configurazione

Il codice sorgente del Voice Assistant può essere ottenuto clonando il repository Git o scaricando l'archivio compresso. Il comando `git clone https://github.com/voice-assistant/navidrome-voice.git` crea una copia locale del repository nella directory corrente. È raccomandato utilizzare Git per facilitare futuri aggiornamenti del sistema.

Una volta scaricato il codice, è necessario creare un ambiente virtuale Python isolato per evitare conflitti con altre applicazioni. Il comando `python3 -m venv venv` crea un nuovo ambiente virtuale nella directory `venv`. L'attivazione dell'ambiente avviene con `source venv/bin/activate` su sistemi Unix/Linux.

L'installazione delle dipendenze Python utilizza pip e il file `requirements.txt` fornito. Il comando `pip install -r requirements.txt` scarica e installa automaticamente tutte le librerie necessarie nelle versioni testate e compatibili. Questo processo può richiedere diversi minuti, specialmente su hardware più lento, poiché alcune dipendenze devono essere compilate dal codice sorgente.

Download dei Modelli

I modelli di machine learning rappresentano la componente più voluminosa dell'installazione e richiedono un download separato. Il sistema include uno script automatizzato `download_models.py` che gestisce il download e la verifica di tutti i modelli necessari.

Lo script scarica automaticamente il modello Vosk italiano raccomandato, i modelli Whisper necessari e i modelli openWakeWord per la detection delle parole di attivazione. Il processo include verifiche di integrità tramite checksum per garantire che i modelli scaricati non siano corrotti.

Per installazioni con banda limitata o requisiti specifici, è possibile scaricare manualmente solo i modelli necessari. Il modello Vosk leggero (vosk-model-small-it-0.22) può essere sufficiente per test iniziali, mentre il modello completo offre accuratezza superiore per uso in produzione.

Configurazione Iniziale

Il sistema utilizza file di configurazione YAML per gestire tutti i parametri operativi. Il file `config/config.example.yaml` fornisce un template completo con valori di default ragionevoli per la maggior parte delle installazioni. Questo file deve essere copiato in `config/config.yaml` e modificato secondo le specifiche esigenze.

La configurazione più critica riguarda i parametri di connessione a Navidrome. È necessario specificare l'URL del server (tipicamente `http://localhost:4533` per installazioni locali), username e password di un account con privilegi di accesso alla libreria musicale. Il sistema supporta anche autenticazione tramite token API per maggiore sicurezza.

I parametri audio richiedono particolare attenzione e potrebbero necessitare di tuning specifico per l'hardware utilizzato. Il sample rate dovrebbe essere impostato a 16000 Hz per compatibilità ottimale con i modelli Vosk, mentre la dimensione del buffer audio può essere regolata per bilanciare latenza e stabilità.

Installazione dei Servizi

Per un funzionamento affidabile in produzione, il Voice Assistant dovrebbe essere configurato come servizio di sistema che si avvia automaticamente al boot. Il sistema include file di configurazione systemd pre-configurati che possono essere installati con privilegi di amministratore.

Il comando `sudo cp systemd/voice-assistant.service /etc/systemd/system/` copia il file di servizio nella directory appropriata. Successivamente, `sudo systemctl daemon-reload` ricarica la configurazione systemd e `sudo systemctl enable voice-assistant` abilita l'avvio automatico del servizio.

Il servizio può essere avviato manualmente con `sudo systemctl start voice-assistant` e il suo stato verificato con

`sudo systemctl status voice-assistant`. I log del servizio sono accessibili tramite `journalctl -u voice-assistant -f` per monitoraggio e debugging.

Verifica dell'Installazione

Una volta completata l'installazione, è importante verificare che tutti i componenti funzionino correttamente. Il sistema include uno script di test

`test_installation.py` che verifica automaticamente la configurazione audio, il caricamento dei modelli, la connettività verso Navidrome e il funzionamento di base del riconoscimento vocale.

Il test audio verifica che il microfono sia accessibile e che i livelli di registrazione siano appropriati. Il test dei modelli carica tutti i modelli di machine learning e verifica che siano compatibili con la versione corrente del software. Il test di connettività verso Navidrome verifica l'autenticazione e l'accesso alla libreria musicale.

Un test completo end-to-end simula un comando vocale completo dalla registrazione audio fino all'esecuzione del comando musicale. Questo test fornisce una misura realistica delle prestazioni del sistema e identifica eventuali problemi di configurazione che potrebbero non essere evidenti nei test individuali dei componenti.

Risoluzione Problemi Comuni

Durante l'installazione possono verificarsi diversi problemi comuni che hanno soluzioni note. Errori di compilazione delle dipendenze Python sono spesso causati da librerie di sviluppo mancanti e possono essere risolti installando i pacchetti `python3-dev`, `libffi-dev` e `libssl-dev`.

Problemi audio sono frequenti su sistemi Linux e possono manifestarsi come impossibilità di accedere al microfono o qualità audio degradata. La verifica dei permessi utente, la configurazione di PulseAudio e l'identificazione del dispositivo audio corretto sono i passi standard per la risoluzione.

Errori di connessione a Navidrome sono tipicamente causati da configurazioni di rete incorrette o credenziali errate. È importante verificare che il server Navidrome sia accessibile dalla macchina che ospita il Voice Assistant e che le credenziali configurate abbiano i privilegi necessari per accedere alla libreria musicale.

Configurazione

La configurazione del Voice Assistant per Navidrome offre un controllo granulare su tutti gli aspetti del sistema, permettendo di ottimizzare le prestazioni per specifici scenari

d'uso e hardware. Il sistema utilizza un approccio a configurazione stratificata che combina valori di default ragionevoli con la possibilità di override dettagliato per utenti avanzati.

Struttura della Configurazione

Il file di configurazione principale utilizza il formato YAML per la sua leggibilità e facilità di modifica. La struttura è organizzata in sezioni logiche che corrispondono ai diversi componenti del sistema: audio, speech recognition, natural language processing, Navidrome integration e system resources.

La sezione audio controlla tutti gli aspetti dell'acquisizione e preprocessing del segnale audio. Il parametro `sample_rate` determina la frequenza di campionamento utilizzata per la registrazione, con 16000 Hz come valore raccomandato per compatibilità ottimale con i modelli Vosk. Valori più alti possono migliorare la qualità audio ma aumentano il carico computazionale e i requisiti di memoria.

Il `buffer_size` controlla la dimensione del buffer audio in samples e rappresenta un compromesso critico tra latenza e stabilità. Buffer più piccoli riducono la latenza ma possono causare dropout audio su sistemi con carico elevato, mentre buffer più grandi aumentano la latenza ma offrono maggiore robustezza. Il valore di default di 1024 samples offre un buon bilanciamento per la maggior parte delle configurazioni.

La configurazione del Voice Activity Detection (VAD) include parametri per la soglia di attivazione, la durata minima del silenzio per considerare terminato un comando vocale, e algoritmi di riduzione del rumore. Il parametro `vad_threshold` controlla la sensibilità del sistema nel distinguere tra voce e rumore di fondo, con valori più bassi che aumentano la sensibilità ma possono causare falsi positivi.

Configurazione Speech Recognition

La sezione speech recognition permette di configurare la strategia ibrida Vosk+Whisper che rappresenta una delle innovazioni chiave del sistema. Il parametro `primary_engine` determina quale motore utilizzare per il riconoscimento iniziale, mentre `fallback_engine` specifica il motore da utilizzare quando il riconoscimento primario non raggiunge la soglia di confidenza minima.

La `confidence_threshold` rappresenta un parametro critico che determina quando il sistema dovrebbe utilizzare il motore di fallback. Valori più alti (0.8-0.9) garantiscono maggiore accuratezza ma aumentano l'utilizzo del motore più lento, mentre valori più bassi (0.6-0.7) privilegiano la velocità a scapito dell'accuratezza in casi ambigui.

Il `fallback_timeout` controlla il tempo massimo che il sistema attende per il completamento del riconoscimento con il motore di fallback. Questo parametro è importante per mantenere una user experience fluida anche quando il riconoscimento richiede elaborazione aggiuntiva.

La configurazione dei modelli include percorsi verso i file dei modelli Vosk e Whisper, permettendo l'utilizzo di modelli personalizzati o ottimizzati per specifici domini. Il sistema supporta anche il caricamento dinamico dei modelli per ridurre l'utilizzo di memoria quando non tutti i motori sono necessari simultaneamente.

Configurazione Natural Language Processing

Il processore di linguaggio naturale offre numerose opzioni di configurazione per ottimizzare l'accuratezza del riconoscimento dei comandi musicali. La sezione `entity_caching` controlla come il sistema mantiene in memoria le informazioni sulla libreria musicale di Navidrome.

Il parametro `cache_refresh_interval` determina la frequenza con cui il sistema aggiorna la cache delle entità musicali. Valori più bassi garantiscono che nuovi contenuti aggiunti a Navidrome siano immediatamente disponibili per i comandi vocali, ma aumentano il carico sul server. L'intervallo di default di 30 minuti rappresenta un buon compromesso per la maggior parte degli utilizzi.

La configurazione del fuzzy matching include parametri per la soglia di similarità utilizzata nel matching di nomi di artisti, album e brani. Il `similarity_threshold` controlla quanto simile deve essere un nome pronunciato rispetto a quello presente nella libreria per essere considerato una corrispondenza valida. Valori più bassi aumentano la tolleranza per errori di pronuncia ma possono causare match incorretti.

Il sistema di context awareness mantiene memoria dei comandi precedenti per migliorare l'interpretazione di comandi ambigui. Il `context_timeout` determina per quanto tempo il sistema ricorda il contesto precedente, bilanciando utilità e privacy (evitando che il sistema mantenga informazioni sui comandi per periodi eccessivamente lunghi).

Configurazione Integrazione Navidrome

La sezione di configurazione per l'integrazione con Navidrome include parametri per ottimizzare le prestazioni di rete e la gestione delle connessioni. Il connection pooling permette di riutilizzare connessioni HTTP esistenti per ridurre l'overhead di stabilimento di nuove connessioni per ogni richiesta.

Il parametro `max_connections` controlla il numero massimo di connessioni simultanee verso il server Navidrome. Valori più alti permettono maggiore parallelismo ma possono sovraccaricare server con risorse limitate. Il valore di default di 10 connessioni è appropriato per la maggior parte delle installazioni domestiche.

La configurazione del caching delle richieste API include parametri per il TTL (Time To Live) dei diversi tipi di dati. I metadati degli artisti e degli album possono essere cachati per periodi più lunghi poiché cambiano raramente, mentre i risultati delle ricerche potrebbero richiedere refresh più frequenti se la libreria viene aggiornata regolarmente.

Il sistema di retry automatico include configurazione per il numero massimo di tentativi, il fattore di backoff esponenziale e i tipi di errori che dovrebbero triggerare un retry. Questa configurazione è particolarmente importante per installazioni su reti wireless o con connettività intermittente.

Configurazione Avanzata del Sistema

La sezione `system resources` permette di ottimizzare l'utilizzo delle risorse hardware per specifiche configurazioni. La gestione della memoria include parametri per limitare l'utilizzo di RAM da parte dei diversi componenti e configurare la frequenza del garbage collection.

Il parametro `memory_limit` imposta un limite massimo per l'utilizzo di memoria da parte del processo principale. Quando questo limite viene raggiunto, il sistema libera automaticamente cache non essenziali e riduce la dimensione dei buffer per rimanere entro i limiti configurati. Questa funzionalità è particolarmente utile su dispositivi con memoria limitata come Raspberry Pi.

La configurazione CPU include opzioni per l'utilizzo di multiprocessing e l'affinità dei processi verso specifici core. Su sistemi multi-core, il sistema può distribuire il carico computazionale tra diversi processi per migliorare le prestazioni complessive. Il parametro `worker_processes` controlla il numero di processi worker utilizzati per l'elaborazione parallela.

Le ottimizzazioni I/O includono configurazioni per l'utilizzo di operazioni asincrone, compressione dei dati in cache e strategie di buffering per l'accesso al disco. Queste ottimizzazioni sono particolarmente importanti su sistemi con storage lento o limitato.

Configurazione di Sicurezza

Anche se il sistema è progettato per funzionare completamente offline, include diverse opzioni di configurazione per la sicurezza. La sezione `security` permette di configurare

l'autenticazione per l'interfaccia web, la crittografia delle comunicazioni interne e le politiche di accesso ai file.

Il parametro `web_authentication` abilita l'autenticazione per l'interfaccia web di controllo, richiedendo username e password per accedere alle funzionalità di configurazione. Questa opzione è raccomandata per installazioni in ambienti condivisi o accessibili da rete.

La configurazione delle comunicazioni interne include opzioni per abilitare la crittografia TLS per le comunicazioni tra componenti distribuiti. Anche se non strettamente necessaria per installazioni monolitiche, questa opzione aggiunge un livello di sicurezza per deployment complessi.

Le politiche di accesso ai file permettono di configurare i permessi per i file di configurazione, log e cache. Il sistema può essere configurato per utilizzare directory specifiche per diversi tipi di dati, facilitando backup selettivi e gestione dei permessi.

Utilizzo del Sistema

L'utilizzo del Voice Assistant per Navidrome è stato progettato per essere intuitivo e naturale, permettendo agli utenti di controllare la propria musica utilizzando comandi vocali in italiano colloquiale. Il sistema comprende una vasta gamma di espressioni e variazioni linguistiche, adattandosi al modo naturale in cui le persone parlano della musica.

Comandi Vocali di Base

I comandi di riproduzione rappresentano la categoria più utilizzata e includono diverse varianti per avviare la musica. Il comando "riproduci" può essere seguito dal nome di un artista ("riproduci Beethoven"), un album ("riproduci l'album Nona Sinfonia"), un brano specifico ("riproduci Moonlight Sonata") o un genere musicale ("riproduci musica classica").

Il sistema comprende anche varianti colloquiali come "metti", "suona", "ascolta" e "avvia", permettendo agli utenti di utilizzare il linguaggio che preferiscono. Frasi come "metti un po' di jazz" o "suona quella canzone di Mozart che mi piace" vengono interpretate correttamente grazie agli algoritmi avanzati di natural language processing.

I comandi di controllo della riproduzione includono "pausa" o "metti in pausa" per interrompere temporaneamente la musica, "riprendi" o "continua" per riprendere la riproduzione, e "stop" o "ferma" per terminare completamente la sessione di ascolto. Il

sistema mantiene lo stato della riproduzione, permettendo di riprendere esattamente dal punto in cui era stata interrotta.

La navigazione tra i brani utilizza comandi come "prossimo", "avanti", "salta" per passare al brano successivo, e "precedente", "indietro", "torna" per tornare al brano precedente. Il sistema supporta anche comandi più specifici come "vai al terzo brano" o "salta i prossimi due brani" per una navigazione più precisa nella coda di riproduzione.

Controllo del Volume e Audio

Il controllo del volume offre diverse modalità di interazione per adattarsi alle preferenze degli utenti. I comandi numerici come "volume al 70%" o "imposta il volume a 50" permettono un controllo preciso del livello audio. Il sistema accetta anche valori in formato percentuale o su scala da 1 a 100.

Per regolazioni relative, il sistema comprende comandi come "alza il volume", "abbassa il volume", "più forte" e "più piano". Questi comandi applicano incrementi predefiniti che possono essere configurati nelle impostazioni del sistema. Comandi più specifici come "alza il volume di 10" permettono controllo granulare degli incrementi.

Il sistema supporta anche comandi per il muting temporaneo con "muto" o "silenzia", e il ripristino dell'audio con "togli il muto" o "riattiva l'audio". Questa funzionalità è particolarmente utile per interruzioni temporanee senza perdere la posizione nella riproduzione.

Gestione di Playlist e Code

La gestione delle playlist rappresenta una delle funzionalità più avanzate del sistema, permettendo di creare, modificare e riprodurre playlist utilizzando solo comandi vocali. Il comando "riproduci playlist" seguito dal nome della playlist avvia la riproduzione di playlist esistenti salvate in Navidrome.

Il sistema permette anche la creazione dinamica di code di riproduzione con comandi come "aggiungi alla coda" seguito dal nome di un brano, artista o album. La coda può essere manipolata con comandi come "rimuovi dalla coda", "sposta in posizione" e "svuota la coda" per un controllo completo della sequenza di riproduzione.

Funzionalità avanzate includono la creazione di playlist intelligenti basate su criteri vocali. Comandi come "crea una playlist con musica simile a questa" o "aggiungi brani dello stesso genere" utilizzano gli algoritmi di raccomandazione per generare automaticamente playlist personalizzate.

Ricerca e Scoperta Musicale

Il sistema di ricerca vocale permette di trovare musica utilizzando criteri diversi e combinazioni complesse. Oltre alla ricerca per nome esatto, il sistema supporta ricerche fuzzy che tollerano errori di pronuncia o variazioni nel nome. Una ricerca per "Betoven" troverà correttamente "Beethoven" grazie agli algoritmi di matching intelligente.

La ricerca può utilizzare criteri multipli combinati in un singolo comando. Frasi come "trova brani di Mozart del periodo classico" o "cerca album di jazz degli anni '60" vengono interpretate correttamente, applicando filtri multipli per restringere i risultati.

Il sistema supporta anche ricerche basate su mood o caratteristiche musicali quando queste informazioni sono disponibili nei metadati. Comandi come "riproduci musica rilassante" o "trova brani energici" utilizzano tag e classificazioni per identificare musica appropriata.

Comandi di Informazione

Gli utenti possono ottenere informazioni sulla riproduzione corrente utilizzando comandi come "che cosa sta suonando?", "chi è l'artista?", "come si chiama questa canzone?" o "da che album è questo brano?". Il sistema fornisce informazioni dettagliate inclusi titolo, artista, album, anno di pubblicazione e durata quando disponibili.

Comandi di stato del sistema come "quanto tempo manca?" forniscono informazioni sul tempo rimanente del brano corrente, mentre "quanti brani ci sono in coda?" informa sulla lunghezza della coda di riproduzione. Il sistema può anche fornire statistiche di ascolto con comandi come "quanto ho ascoltato oggi?" quando il tracking è abilitato.

Modalità Avanzate

Il sistema supporta diverse modalità di riproduzione che possono essere attivate vocalmente. La modalità shuffle può essere attivata con "attiva shuffle", "modalità casuale" o "mescola i brani", mentre la modalità repeat utilizza comandi come "ripeti" o "modalità ripetizione".

La modalità party attiva funzionalità specifiche per feste o eventi, come volume automatico più alto, preferenza per brani con alto rating e disabilitazione di pause automatiche. Questa modalità può essere attivata con "modalità festa" o "attiva party mode".

Per utilizzo notturno, la modalità sleep riduce gradualmente il volume e si spegne automaticamente dopo un tempo configurabile. Comandi come "modalità notte" o "spegniti tra 30 minuti" attivano questa funzionalità con timer personalizzabili.

Personalizzazione e Apprendimento

Il sistema include funzionalità di apprendimento che migliorano l'accuratezza del riconoscimento nel tempo. Quando un comando non viene riconosciuto correttamente, gli utenti possono fornire feedback con "no, intendevo..." seguito dalla correzione, permettendo al sistema di apprendere dalle preferenze individuali.

Le preferenze musicali vengono apprese automaticamente attraverso l'utilizzo, con il sistema che nota quali artisti, generi e brani vengono richiesti più frequentemente. Queste informazioni vengono utilizzate per migliorare i suggerimenti e dare priorità ai risultati di ricerca più rilevanti per l'utente specifico.

Il sistema permette anche la definizione di comandi personalizzati attraverso l'interfaccia di configurazione. Gli utenti possono creare alias per comandi complessi o definire scorciatoie per le loro playlist preferite, rendendo l'interazione ancora più naturale e personalizzata.

Interfaccia Web

L'interfaccia web del Voice Assistant per Navidrome fornisce un centro di controllo completo per la gestione, configurazione e monitoraggio del sistema. Sviluppata utilizzando React e tecnologie web moderne, l'interfaccia offre un'esperienza utente fluida e responsive che si adatta automaticamente a dispositivi desktop, tablet e smartphone.

Dashboard Principale

La dashboard principale rappresenta il punto di accesso centrale per tutte le funzionalità del sistema e fornisce una panoramica in tempo reale dello stato operativo. La sezione di stato del sistema mostra indicatori visivi per ogni componente principale: Audio Input Manager, Wake Word Detection, Speech Recognition, NLP Processor e Music Controller.

Ogni componente è rappresentato da un badge colorato che indica lo stato corrente: verde per operativo, giallo per warning e rosso per errori. Cliccando su ogni badge, gli utenti possono accedere a informazioni dettagliate inclusi metriche di performance, log degli ultimi eventi e opzioni di riavvio del componente specifico.

La sezione di riproduzione corrente mostra informazioni dettagliate sul brano in corso, inclusi titolo, artista, album, artwork quando disponibile, e una barra di progresso interattiva. Gli utenti possono controllare la riproduzione direttamente dall'interfaccia web utilizzando i controlli standard play/pause, previous/next e regolazione del volume.

La cronologia dei comandi recenti fornisce una vista in tempo reale degli ultimi comandi vocali processati dal sistema. Ogni voce include il testo riconosciuto, il timestamp, lo stato di esecuzione (successo/errore) e il tempo di elaborazione. Questa informazione è preziosa per debugging e per comprendere come il sistema interpreta i comandi vocali.

Sezione Controlli

La sezione controlli offre interfacce grafiche per tutte le funzionalità principali del sistema, permettendo controllo manuale quando i comandi vocali non sono disponibili o pratici. Il pannello di controllo del riconoscimento vocale include un grande pulsante per attivare/disattivare l'ascolto, indicatori del livello audio in tempo reale e controlli per la calibrazione del microfono.

Il visualizzatore del livello audio mostra un grafico in tempo reale dell'input del microfono, permettendo agli utenti di verificare che il sistema stia ricevendo audio correttamente e di identificare problemi come livelli troppo bassi o distorsione. Il sistema include anche un test del microfono che registra un breve campione audio e lo riproduce per verificare la qualità dell'acquisizione.

I controlli di riproduzione musicale replicano tutte le funzionalità disponibili tramite comandi vocali in un'interfaccia grafica intuitiva. Oltre ai controlli standard, l'interfaccia include funzionalità avanzate come la gestione della coda di riproduzione con drag-and-drop per riordinare i brani, ricerca in tempo reale nella libreria musicale e creazione di playlist temporanee.

Il pannello di controllo del volume include uno slider principale per il volume generale e controlli separati per diversi canali audio quando supportati dall'hardware. Il sistema supporta anche preset di volume per diverse situazioni (giorno, notte, festa) che possono essere attivati con un singolo click.

Configurazione del Sistema

La sezione di configurazione fornisce accesso a tutti i parametri del sistema attraverso un'interfaccia user-friendly che nasconde la complessità dei file di configurazione sottostanti. La configurazione è organizzata in schede tematiche che corrispondono ai diversi aspetti del sistema.

La scheda Audio permette di configurare tutti i parametri relativi all'acquisizione e elaborazione dell'audio. Gli utenti possono selezionare il dispositivo di input da un menu dropdown che elenca tutti i microfoni disponibili, regolare il sample rate e la dimensione del buffer, e configurare algoritmi di riduzione del rumore. L'interfaccia include anche un test automatico che verifica la configurazione e suggerisce ottimizzazioni basate sull'hardware rilevato.

La configurazione del riconoscimento vocale include opzioni per selezionare i modelli da utilizzare, regolare le soglie di confidenza e configurare la strategia ibrida Vosk+Whisper. L'interfaccia mostra informazioni dettagliate su ogni modello disponibile, incluse dimensioni, accuratezza stimata e requisiti computazionali, aiutando gli utenti a fare scelte informate.

La sezione di configurazione Navidrome permette di inserire le credenziali del server, testare la connessione e configurare parametri avanzati come connection pooling e caching. L'interfaccia include un wizard di configurazione guidata che automatizza il processo di discovery del server e verifica della connettività.

Monitoraggio e Statistiche

La sezione di monitoraggio fornisce insight dettagliati sulle prestazioni del sistema e sull'utilizzo nel tempo. I grafici di performance mostrano metriche chiave come latenza media dei comandi, accuratezza del riconoscimento vocale, utilizzo di CPU e memoria, e throughput delle richieste.

Le statistiche di utilizzo includono informazioni sui comandi più utilizzati, artisti e generi più richiesti, e pattern di utilizzo nel tempo. Questi dati aiutano gli utenti a comprendere come utilizzano il sistema e possono informare decisioni di ottimizzazione della configurazione.

Il sistema di alerting integrato monitora continuamente le metriche di sistema e può inviare notifiche quando vengono rilevate anomalie. Gli alert possono essere configurati per diversi tipi di eventi: errori di sistema, degradazione delle prestazioni, problemi di connettività con Navidrome, e soglie di utilizzo delle risorse.

La sezione diagnostica include strumenti per il troubleshooting avanzato, inclusi viewer dei log in tempo reale, test di connettività di rete, e benchmark delle prestazioni. Questi strumenti sono particolarmente utili per identificare e risolvere problemi di configurazione o performance.

Gestione Utenti e Sicurezza

L'interfaccia web include un sistema di gestione utenti che permette di creare account separati per diversi membri della famiglia o utenti del sistema. Ogni utente può avere preferenze personalizzate, cronologia dei comandi separata e livelli di accesso differenziati alle funzionalità di configurazione.

Il sistema di autenticazione supporta diverse modalità: autenticazione locale con username/password, integrazione con sistemi di autenticazione esistenti tramite LDAP o

OAuth, e autenticazione a due fattori per maggiore sicurezza. Le sessioni utente sono gestite in modo sicuro con token JWT e scadenza automatica configurabile.

I controlli di accesso permettono di limitare l'accesso a specifiche funzionalità basandosi sui ruoli utente. Gli amministratori hanno accesso completo a tutte le funzionalità, mentre gli utenti standard possono essere limitati al controllo della riproduzione senza accesso alle configurazioni di sistema.

Responsive Design e Accessibilità

L'interfaccia è stata progettata seguendo principi di responsive design per garantire un'esperienza ottimale su tutti i dispositivi. Su smartphone, l'interfaccia si adatta automaticamente con menu collassabili, controlli touch-friendly e layout ottimizzato per schermi piccoli.

Il sistema include funzionalità di accessibilità complete per utenti con disabilità. Tutti i controlli sono accessibili tramite tastiera, l'interfaccia è compatibile con screen reader, e include opzioni per alto contrasto e dimensioni del testo aumentate. I controlli vocali del sistema rappresentano essi stessi una funzionalità di accessibilità importante per utenti con limitazioni motorie.

L'interfaccia supporta anche modalità offline limitata che permette di visualizzare lo stato del sistema e controllare la riproduzione anche quando la connessione di rete è intermittente. Le modifiche alla configurazione vengono salvate localmente e sincronizzate automaticamente quando la connettività viene ripristinata.

Riferimenti

[1] Amazon Alexa Privacy Documentation. "How Alexa Works with Your Personal Information." Amazon Web Services. <https://www.amazon.com/gp/help/customer/display.html?nodeId=GA7E98TJFEJLYSFR>

[2] Povey, Daniel, et al. "The Kaldi Speech Recognition Toolkit." IEEE 2011 Workshop on Automatic Speech Recognition and Understanding. <https://kaldi-asr.org/>

[3] Subsonic API Documentation. "Subsonic API Reference." Subsonic.org. <http://www.subsonic.org/pages/api.jsp>

[4] Ravanelli, Mirco, et al. "SpeechBrain: A General-Purpose Speech Toolkit." arXiv preprint arXiv:2106.04624 (2021). <https://arxiv.org/abs/2106.04624>

[5] Radford, Alec, et al. "Robust Speech Recognition via Large-Scale Weak Supervision." arXiv preprint arXiv:2212.04356 (2022). <https://arxiv.org/abs/2212.04356>

[6] Vosk Speech Recognition Toolkit. "Vosk API Documentation." Alpha Cephei. <https://alphacephei.com/vosk/>

[7] OpenWakeWord Documentation. "Customizable Wake Word Detection." GitHub. <https://github.com/dscripka/openWakeWord>

[8] Navidrome Documentation. "Modern Music Server and Streamer." Navidrome.org. <https://www.navidrome.org/>

[9] FastAPI Documentation. "FastAPI Framework for Building APIs." FastAPI. <https://fastapi.tiangolo.com/>

[10] Redis Documentation. "In-Memory Data Structure Store." Redis.io. <https://redis.io/documentation>

Appendici

Appendice A: Comandi Vocali Supportati

Comandi di Riproduzione

- "Riproduci [artista/album/brano]"
- "Metti [contenuto musicale]"
- "Suona [contenuto musicale]"
- "Ascolta [contenuto musicale]"
- "Avvia [contenuto musicale]"

Comandi di Controllo

- "Pausa" / "Metti in pausa"
- "Riprendi" / "Continua"
- "Stop" / "Ferma"
- "Prossimo" / "Avanti" / "Salta"
- "Precedente" / "Indietro" / "Torna"

Comandi Volume

- "Volume al [numero]%"
- "Imposta volume a [numero]"
- "Alza il volume"
- "Abbassa il volume"
- "Più forte" / "Più piano"

- "Muto" / "Silenzia"

Comandi Informazione

- "Che cosa sta suonando?"
- "Chi è l'artista?"
- "Come si chiama questa canzone?"
- "Da che album è questo brano?"

Appendice B: Codici di Errore

Errori Audio (1000-1099)

- 1001: Microfono non accessibile
- 1002: Formato audio non supportato
- 1003: Buffer audio overflow
- 1004: Dispositivo audio disconnesso

Errori Speech Recognition (1100-1199)

- 1101: Modello Vosk non caricato
- 1102: Modello Whisper non disponibile
- 1103: Timeout riconoscimento vocale
- 1104: Confidenza sotto soglia minima

Errori NLP (1200-1299)

- 1201: Comando non riconosciuto
- 1202: Entità musicale non trovata
- 1203: Comando ambiguo
- 1204: Cache entità non disponibile

Errori Navidrome (1300-1399)

- 1301: Connessione server fallita
- 1302: Autenticazione non riuscita
- 1303: Risorsa non trovata
- 1304: Timeout richiesta API

Appendice C: File di Configurazione Esempio

```
# Configurazione Voice Assistant per Navidrome  
# Versione: 1.0
```

audio:

```
# Configurazione acquisizione audio
sample_rate: 16000
channels: 1
buffer_size: 1024
device_index: null # Auto-detect

# Voice Activity Detection
vad_threshold: 0.3
silence_duration: 1.0
noise_reduction: true
```

speech_recognition:

```
# Strategia ibrida
primary_engine: "vosk"
fallback_engine: "whisper"
confidence_threshold: 0.7
fallback_timeout: 2.0

# Modelli
vosk_model: "models/vosk-model-it-0.22"
whisper_model: "medium"

# Cache
enable_cache: true
cache_size: 1000
cache_ttl: 3600
```

nlp:

```
# Elaborazione linguaggio naturale
language: "it"
fuzzy_threshold: 0.6
context_timeout: 300

# Cache entità
entity_cache_size: 5000
entity_refresh_interval: 1800
```

navidrome:

```
# Connessione server
base_url: "http://localhost:4533"
username: "your_username"
password: "your_password"

# Ottimizzazioni
connection_pool_size: 10
request_timeout: 10
cache_ttl: 900
```

system:

```
# Risorse sistema
memory_limit: "1GB"
```



```
cpu_cores: null # Auto-detect
log_level: "INFO"

# Sicurezza
web_auth_enabled: false
api_key: null
```

Appendice D: Script di Installazione Automatica

```
#!/bin/bash
# Script di installazione automatica Voice Assistant per
# Navidrome
# Versione: 1.0

set -e

echo "🎵 Voice Assistant per Navidrome - Installazione
Automatica"
echo
"===== "

# Verifica sistema operativo
if [[ "$OSTYPE" != "linux-gnu"* ]]; then
    echo
    "❌ Errore: Sistema operativo non supportato. Richiesto Linux."
    exit 1
fi


# Aggiornamento sistema
echo "📦 Aggiornamento pacchetti sistema..."
sudo apt update && sudo apt upgrade -y


# Installazione dipendenze sistema
echo "🔧 Installazione dipendenze..."
sudo apt install -y \
    python3 python3-pip python3-venv \
    build-essential python3-dev \
    portaudio19-dev alsa-utils \
    pulseaudio pulseaudio-utils \
    git curl wget redis-server


# Creazione directory progetto
PROJECT_DIR="$HOME/voice-assistant"
echo "📁 Creazione directory progetto: $PROJECT_DIR"
mkdir -p "$PROJECT_DIR"
cd "$PROJECT_DIR"


# Download codice sorgente
echo "⬇️ Download codice sorgente..."
git clone https://github.com/voice-assistant/navidrome-
```


```
voice.git .


# Creazione ambiente virtuale
echo " Creazione ambiente virtuale Python..."
python3 -m venv venv
source venv/bin/activate




# Installazione dipendenze Python
echo " Installazione dipendenze Python..."
pip install --upgrade pip
pip install -r requirements.txt

# Download modelli
echo " Download modelli machine learning..."
python scripts/download_models.py

# Configurazione iniziale
echo " Configurazione iniziale..."
cp config/config.example.yaml config/config.yaml

# Test installazione
echo " Test installazione..."
python scripts/test_installation.py

# Installazione servizio systemd
echo " Installazione servizio sistema..."
sudo cp systemd/voice-assistant.service /etc/systemd/system/
sudo systemctl daemon-reload
sudo systemctl enable voice-assistant

echo " Installazione completata!"
echo ""
echo " Prossimi passi:"
echo "1. Modifica config/config.yaml con le tue impostazioni Navidrome"
echo "2. Avvia il servizio: sudo systemctl start voice-assistant"
echo "3. Accedi all'interfaccia web: http://localhost:8080"
echo ""
echo " Documentazione completa: docs/voice_assistant_documentation.pdf"
```

Questo documento rappresenta la guida completa per l'installazione, configurazione e utilizzo del Voice Assistant per Navidrome. Per supporto tecnico o contributi al progetto, visitare il repository GitHub ufficiale.

Autore: Manus AI

Licenza: MIT License

Versione Documento: 1.0

Data Ultima Modifica: 15 Giugno 2025