ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Dipartimento di Informatica – Scienza e Ingegneria

Corso di Laurea in Informatica

Whisper per la trascrizione automatica in ambito agricolo: caso di studio e analisi critica dello stato dell'arte

Relatore:
Chiar.mo Prof.
Andrea Asperti

Correlatore:
Marco Miana

Presentata da:
Francesco Ricci

Sessione I
Anno Accademico 2025/2026

Parole chiave

adattamento di dominio

trascrizione automatica

agricoltura

Whisper

Automatic Speech Recognition

Abstract

Questa tesi analizza l'utilizzo del modello Whisper di OpenAI, accessibile tramite API in modalità *zero-shot*, per la trascrizione automatica di attività agricole. Il lavoro inquadra il caso di studio *Traccial'Abilità* nello stato dell'arte dei sistemi di *Automatic Speech Recognition*, con particolare attenzione ai limiti dei modelli generalisti nel riconoscimento di lessico tecnico, termini rari e nomi commerciali. Un principio centrale del sistema analizzato è che la trascrizione non sostituisce mai la registrazione: l'audio resta la fonte primaria dell'informazione, mentre il testo svolge una funzione di supporto rivedibile. L'analisi evidenzia i vantaggi dell'uso tramite API, ne discute i limiti — dal lessico di dominio alle allucinazioni — e valuta quali tecniche leggere di adattamento siano realisticamente trasferibili a uno scenario basato su API.

Indice

Elenco delle tabelle

Elenco delle figure

Capitolo 0

INTRODUZIONE

Negli ultimi anni, i sistemi di *Automatic Speech Recognition* (ASR) hanno conosciuto un significativo avanzamento grazie all'introduzione di modelli neurali di grandi dimensioni addestrati su ampie raccolte di dati. In questo contesto, Whisper, proposto da Radford et al.[1], rappresenta uno dei modelli più rilevanti, grazie alla sua capacità di operare in modalità *zero-shot* e alla sua robustezza rispetto a diversi domini applicativi.

La presente tesi si inserisce in questo scenario con l'obiettivo di analizzare l'utilizzo del modello Whisper, tramite API OpenAI, all'interno di un contesto reale: la trascrizione vocale delle attività svolte da operatori agricoli durante il lavoro in campo.

Il caso applicativo considerato è *Traccial'Abilità*, un sistema pensato per semplificare la raccolta e l'elaborazione delle informazioni necessarie alla compilazione del Quaderno di Campagna dell'Agricoltore (QDCA). Il QDCA richiede la dichiarazione dettagliata delle operazioni svolte sulle colture e dei prodotti utilizzati durante l'anno. In un contesto in cui tale documentazione è richiesta in formato digitale, la possibilità di registrare informazioni operative tramite voce rappresenta un supporto rilevante per ridurre il carico manuale sull'operatore.

Un aspetto centrale del sistema è che la trascrizione non sostituisce mai l'audio originale. Ogni trascrizione viene salvata insieme alla registrazione da cui è stata generata: la fonte primaria dell'informazione rimane quindi l'audio prodotto dall'operatore agricolo. La trascrizione ha invece una funzione strumentale: rende più semplice consultare, elaborare e trasformare il contenuto registrato in informazioni utili alla gestione del QDCA, mantenendo al tempo stesso la possibilità di risalire sempre alla fonte originaria.

Questo contesto presenta specifiche criticità, tra cui l'utilizzo di terminologia tecnica, nomi commerciali di prodotti, riferimenti a colture, trattamenti e condizioni ambientali non controllate. L'obiettivo della tesi è quindi quello di valutare in che misura l'utilizzo di Whisper tramite API, senza tecniche avanzate di adattamento, sia coerente con lo stato dell'arte in letteratura, con particolare riferimento ai lavori che estendono o analizzano il paper di Radford.

La trattazione procede dai fondamenti tecnici verso il caso applicativo: dopo aver richiamato i principi generali dei sistemi di *Automatic Speech Recognition* e le caratteristiche specifiche di Whisper e del suo utilizzo tramite API, si ripercorre la letteratura sull'adattamento di dominio, sul lessico tecnico e sui limiti dei modelli generalisti, per poi descrivere il sistema *Traccial'Abilità* e discuterne criticamente le scelte progettuali alla luce dello stato dell'arte.

Capitolo 1

Automatic Speech Recognition

L'*Automatic Speech Recognition* (ASR), o riconoscimento automatico del parlato, è l'insieme delle tecniche informatiche e statistiche finalizzate alla conversione di un segnale vocale in una rappresentazione testuale. In termini generali, un sistema ASR riceve in input un file audio o un flusso audio continuo e produce in output una trascrizione, eventualmente arricchita da informazioni aggiuntive come punteggiatura, segmentazione temporale, identificazione della lingua o riconoscimento degli interlocutori.

L'ASR rappresenta oggi una componente fondamentale di numerose applicazioni: assistenti vocali, sistemi di sottotitolazione automatica, trascrizione di riunioni, strumenti di accessibilità, interfacce uomo-macchina e applicazioni verticali in ambito medico, legale, industriale e agricolo. Nel caso della presente tesi, l'ASR viene considerato come tecnologia abilitante per affiancare alle annotazioni vocali raccolte in campo una rappresentazione testuale più facilmente consultabile, elaborabile e integrabile in un sistema informativo aziendale. La trascrizione, tuttavia, non viene intesa come sostituto della registrazione: l'audio originale rimane la fonte primaria dell'informazione, mentre il testo trascritto svolge una funzione di supporto operativo.

1.1 Obiettivo e funzionamento generale di un sistema ASR

Il problema affrontato da un sistema ASR può essere formulato come la ricerca della sequenza di parole più probabile dato un segnale audio osservato. Il segnale vocale è una forma d'onda continua, variabile nel tempo, influenzata da molteplici fattori:

timbro del parlante, velocità di pronuncia, accento, rumore di fondo, qualità del microfono e condizioni ambientali.

Un sistema ASR deve quindi risolvere contemporaneamente più problemi:

- trasformare il segnale audio in una rappresentazione numerica utilizzabile dal modello;
- distinguere il parlato da rumore, silenzio o suoni non linguistici;
- riconoscere fonemi, sillabe o unità linguistiche;
- ricostruire parole e frasi coerenti;
- produrre un testo leggibile e utile per l'utente.

Questa complessità rende l'ASR un problema diverso dalla semplice classificazione di suoni. Il sistema non deve soltanto riconoscere pattern acustici, ma anche interpretarli alla luce di una lingua, un vocabolario e un contesto.

Nel caso di applicazioni reali, inoltre, il modello non lavora quasi mai su audio perfetti: la registrazione può avvenire in presenza di rumore ambientale, sovrapposizioni o interruzioni. La qualità del riconoscimento dipende quindi non solo dalla potenza del modello, ma anche dalla sua robustezza rispetto a condizioni non controllate.

1.2 Dal segnale vocale alla rappresentazione numerica

Il primo passaggio di un sistema ASR consiste nella trasformazione del segnale audio grezzo in una rappresentazione adatta all'elaborazione automatica. Il parlato viene normalmente campionato a una certa frequenza e suddiviso in finestre temporali brevi, poiché le caratteristiche acustiche del segnale variano rapidamente nel tempo.

Storicamente, una delle rappresentazioni più utilizzate è costituita dai coefficienti MFCC (*Mel-Frequency Cepstral Coefficients*), progettati per approssimare alcune proprietà della percezione uditiva umana. Nei sistemi più recenti, e in particolare nei modelli neurali *end-to-end*, è frequente l'utilizzo di spettrogrammi *log-Mel* (Figure 1.1), che rappresentano l'evoluzione temporale dell'energia del segnale a diverse frequenze.

Questa trasformazione è importante perché consente al modello di lavorare su una rappresentazione più compatta e informativa rispetto alla forma d'onda grezza. Anche Whisper, come molti modelli moderni, utilizza una rappresentazione au-

dio basata su *log-Mel spectrogram* prima di applicare l'architettura neurale vera e propria.

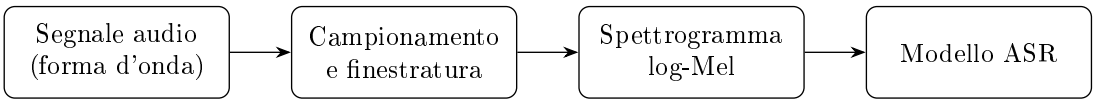

Figura 1.1: Schematizzazione del *front-end* di un sistema ASR: la forma d'onda viene campionata e suddivisa in finestre temporali brevi, dalle quali si ricava una rappresentazione tempo–frequenza — qui uno spettrogramma *log-Mel* — fornita in ingresso al modello.

1.3 Sistemi ASR tradizionali

I primi sistemi ASR erano generalmente composti da più moduli separati, ciascuno responsabile di una parte specifica del processo. L'architettura tradizionale prevedeva tipicamente:

- un modello acustico, incaricato di associare porzioni del segnale audio a unità fonetiche;
- un dizionario di pronuncia, che collegava le parole alle sequenze di fonemi;
- un modello linguistico, utilizzato per stimare la probabilità delle sequenze di parole;
- un algoritmo di *decoding*, incaricato di combinare le informazioni acustiche e linguistiche per produrre la trascrizione finale.

In questo paradigma, modelli come gli *Hidden Markov Models* (HMM) hanno avuto un ruolo centrale. Gli HMM consentivano di modellare la sequenza temporale del parlato, rappresentando il processo di generazione del segnale come una successione di stati nascosti. Spesso venivano combinati con *Gaussian Mixture Models* (GMM) per modellare la distribuzione delle caratteristiche acustiche.

Questi sistemi hanno avuto un grande successo, ma presentavano diversi limiti. In primo luogo, richiedevano una progettazione manuale complessa, con componenti separati da ottimizzare individualmente. In secondo luogo, dipendevano fortemente dalla qualità del dizionario di pronuncia e del modello linguistico. Infine, tendevano a essere poco robusti quando applicati a domini diversi da quelli per cui erano stati progettati o addestrati.

1.4 L'impatto del deep learning

L'introduzione del *deep learning* ha modificato profondamente l'evoluzione dei sistemi ASR. Le reti neurali profonde hanno permesso di sostituire progressivamente i modelli acustici tradizionali con modelli capaci di apprendere rappresentazioni più ricche direttamente dai dati.

In una prima fase, le reti neurali sono state integrate nei sistemi esistenti, sostituendo ad esempio i GMM nei modelli HMM-GMM. Successivamente, l'evoluzione delle architetture neurali ha portato allo sviluppo di modelli sempre più *end-to-end*, in grado di apprendere direttamente la relazione tra input audio e output testuale.

Tra le architetture più rilevanti si possono citare:

- modelli basati su *Connectionist Temporal Classification* (CTC), utili per allineare sequenze audio e trascrizioni senza una segmentazione manuale precisa;
- modelli *sequence-to-sequence* con attenzione, capaci di generare testo in modo autoregressivo;
- modelli Transformer, che hanno migliorato la capacità di modellare dipendenze a lungo raggio e di scalare su grandi quantità di dati.

Il passaggio ai modelli *end-to-end* ha ridotto la necessità di componenti progettate manualmente, permettendo al modello di apprendere internamente molte delle trasformazioni prima gestite separatamente. Tuttavia, questa semplificazione architetturale ha reso ancora più centrale la disponibilità di grandi quantità di dati di addestramento.

1.5 Pre-training, grandi dataset e modelli foundation

Un elemento decisivo nello sviluppo recente dell'ASR è rappresentato dal *pre-training* su larga scala. L'idea generale è addestrare un modello su grandi quantità di dati, in modo che apprenda rappresentazioni generali del parlato e della lingua, riutilizzabili poi in molteplici contesti applicativi.

La ricerca recente ha seguito due direzioni principali:

- approcci *self-supervised* o *semi-supervised*, che sfruttano grandi quantità di audio non trascritto;
- approcci *weakly supervised*, che utilizzano grandi quantità di audio associato a trascrizioni non necessariamente perfette.

Whisper appartiene a questa seconda categoria. Il modello è stato addestrato su un corpus molto ampio e diversificato, con l'obiettivo di ottenere un sistema robusto e generalizzabile. La scelta di utilizzare dati debolmente supervisionati su larga scala risponde a una necessità pratica: i dataset perfettamente annotati sono costosi e limitati, mentre grandi quantità di audio con trascrizioni imperfette sono più facilmente reperibili.

Questa impostazione ha portato alla nascita di modelli ASR assimilabili ai cosiddetti *foundation models*: sistemi addestrati su larga scala, non specializzati su un singolo dominio, ma utilizzabili in molteplici scenari applicativi. La loro forza non consiste necessariamente nell'essere ottimizzati per un compito ristretto, ma nell'offrire buone prestazioni generali anche in assenza di adattamento specifico.

1.6 Zero-shot e adattamento di dominio

Uno dei concetti centrali per comprendere Whisper è quello di *zero-shot learning*. In ambito ASR, un modello opera in modalità *zero-shot* quando viene utilizzato su un dominio o un dataset senza essere stato esplicitamente addestrato o ottimizzato su quel contesto specifico.

Questo aspetto è particolarmente importante per applicazioni industriali e aziendali, poiché addestrare o adattare un modello richiede:

- dati audio rappresentativi del dominio;
- trascrizioni corrette e validate;
- risorse computazionali;
- competenze specialistiche;
- procedure di valutazione e manutenzione.

In molti progetti reali, questi requisiti non sono sostenibili. L'utilizzo di un modello *zero-shot* tramite API, invece, consente di integrare rapidamente una funzionalità di trascrizione automatica, accettando alcuni compromessi.

Il principale limite dello *zero-shot* riguarda l'adattamento al dominio. Un modello generalista può trascrivere correttamente la maggior parte del parlato comune, ma può incontrare difficoltà con parole rare, abbreviazioni, nomi propri o terminologia specialistica, categorie in cui rientra anche il lessico tecnico dei domini verticali.

1.7 Metriche di valutazione

La valutazione di un sistema ASR viene comunemente effettuata confrontando la trascrizione prodotta dal modello con una trascrizione di riferimento. La metrica più utilizzata è il *Word Error Rate* (WER), che misura il numero di errori necessari per trasformare l'output del sistema nella trascrizione corretta.

Gli errori considerati dal WER sono:

- sostituzioni, quando una parola viene riconosciuta come un'altra;
- cancellazioni, quando una parola presente nell'audio non compare nella trascrizione;
- inserzioni, quando il modello aggiunge parole non presenti nell'audio.

Formalmente, il WER è definito come

$$\mathrm{WER} = \frac{S + D + I}{N},$$

dove S, D e I indicano rispettivamente il numero di sostituzioni, cancellazioni e inserzioni, mentre N è il numero totale di parole nella trascrizione di riferimento.

Il WER è utile perché fornisce una misura quantitativa semplice, ma presenta anche alcuni limiti. Penalizza infatti differenze che possono essere poco rilevanti per l'utente, come variazioni di punteggiatura, maiuscole, forma dei numeri o piccole differenze stilistiche. Per questo motivo, nei sistemi moderni si utilizzano spesso procedure di normalizzazione del testo prima del calcolo della metrica.

In alcuni casi viene utilizzato anche il *Character Error Rate* (CER), particolarmente utile per lingue o contesti in cui la segmentazione in parole è meno stabile. Va osservato che la valutazione puramente quantitativa non è sempre sufficiente: non tutti gli errori hanno lo stesso peso applicativo, poiché un errore su una parola comune può risultare trascurabile, mentre un errore su un termine critico può compromettere l'elaborazione automatica del testo.

1.8 Sfide dell'ASR in contesti reali

L'applicazione dell'ASR in contesti reali introduce una serie di difficoltà che spesso non emergono pienamente nei benchmark standard. Tra le principali sfide si possono individuare:

- rumore ambientale;
- parlato spontaneo e non pianificato;

- accenti e variabilità individuale;
- sovrapposizione tra parlato e rumori esterni;
- termini tecnici e vocabolario specialistico;
- necessità di trascrizioni leggibili e correggibili dall'utente.

Questi fattori si presentano spesso in combinazione e ne amplificano l'impatto: l'utente può parlare in modo rapido, interrompersi, utilizzare abbreviazioni o riferirsi a elementi noti nel contesto d'uso ma non esplicitamente presenti nell'audio. L'obiettivo, in scenari di questo tipo, non è soltanto ottenere una trascrizione grammaticalmente corretta, ma produrre un testo utile, affidabile e coerente con ciò che è stato effettivamente detto.

1.9 Rilevanza per la presente tesi

Alla luce di quanto esposto, l'ASR non deve essere considerato soltanto come una tecnologia di conversione audio-testo, ma come una componente di un sistema informativo più ampio. Nel progetto *Traccial'Abilità*, la trascrizione automatica ha il compito di ridurre lo sforzo dell'utente, facilitare la registrazione delle attività e rendere più accessibili e certe le informazioni raccolte sul campo.

Il sistema si inserisce nel processo di gestione del Quaderno di Campagna dell'Agricoltore, dove è necessario documentare in modo puntuale le operazioni effettuate sulle colture e i prodotti impiegati. In questo contesto, la registrazione vocale consente all'operatore di dichiarare l'attività nel momento in cui viene svolta o in prossimità di essa, riducendo il rischio di perdita dell'informazione. La trascrizione automatica permette poi di rendere tale contenuto più facilmente elaborabile, ma senza sostituire l'audio originale, che rimane associato al testo e costituisce il riferimento primario in caso di ambiguità o correzione. L'informazione registrata diventa poi elemento di collaborazione e verifica tra gli attori di filiera.

L'utilizzo di Whisper tramite API rappresenta quindi una scelta progettuale pragmatica: adottare un modello generalista robusto, evitando i costi e la complessità del *fine-tuning*. Questa scelta è coerente con l'evoluzione recente dei modelli ASR, ma apre anche questioni critiche legate all'adattamento di dominio, alla gestione del lessico tecnico e all'affidabilità della trascrizione come strumento di supporto, non come fonte autonoma dell'informazione.

Proprio per questo motivo, Whisper rappresenta un caso particolarmente rilevante per la presente tesi. A differenza di soluzioni ASR che richiedono una fase di addestramento o adattamento su dati specifici, Whisper può essere integrato diret-

tamente in un sistema applicativo tramite API, rendendo possibile un uso pratico dell'ASR anche in assenza di un dataset agricolo annotato o di infrastrutture dedicate al training. Il suo interesse non deriva quindi soltanto dalle prestazioni del modello, ma anche dalla possibilità di trasferire rapidamente una tecnologia avanzata in un contesto operativo reale.

Il capitolo successivo approfondisce quindi il modello Whisper, analizzandone architettura, modalità di addestramento, capacità *zero-shot* e limiti. Questi elementi sono necessari per comprendere sia l'adeguatezza del modello a un'integrazione pragmatica come quella realizzata in *Traccial'Abilità*, sia il permanere, in presenza di terminologia agricola specialistica, di problemi di adattamento di dominio discussi nei capitoli successivi.

Capitolo 2

Il modello Whisper

Whisper è un modello di *Automatic Speech Recognition* sviluppato da OpenAI e presentato da Radford et al. nel lavoro *Robust Speech Recognition via Large-Scale Weak Supervision*[1]. Il modello si colloca nel contesto dei sistemi ASR moderni basati su reti neurali profonde e, in particolare, su architetture Transformer *encoder–decoder*. La sua rilevanza non dipende soltanto dalle prestazioni ottenute su specifici benchmark, ma soprattutto dall'impostazione progettuale adottata: Whisper è stato addestrato per funzionare in modo robusto su una grande varietà di domini, lingue e condizioni audio, senza richiedere necessariamente *fine-tuning* sul dominio di destinazione.

Questa caratteristica lo rende particolarmente interessante per applicazioni pratiche come quella analizzata nella presente tesi. Nel caso di *Traccial'Abilità*, il modello viene utilizzato tramite API, senza accesso diretto ai pesi e senza una fase di addestramento specifica su dati agricoli. Whisper viene quindi adottato come componente generalista di trascrizione automatica, integrata all'interno di un sistema più ampio in cui la registrazione audio rimane la fonte primaria dell'informazione e la trascrizione svolge una funzione di supporto all'elaborazione.

2.1 Obiettivo del modello

L'obiettivo dichiarato nel lavoro originale è sviluppare un sistema di riconoscimento del parlato capace di generalizzare bene a contesti diversi da quelli specificamente usati per la valutazione. Gli autori osservano che molti sistemi ASR raggiungono prestazioni molto elevate su dataset standard, ma possono degradare sensibilmente quando vengono applicati a dati fuori distribuzione, cioè diversi per parlanti, rumore, dominio, lingua o stile del parlato.

Il problema non è quindi soltanto ottenere una bassa percentuale di errore su un singolo benchmark, ma costruire un modello che funzioni in modo affidabile *out of the box*, cioè senza dover essere adattato ogni volta a un nuovo contesto. Questa impostazione è centrale per comprendere Whisper: il modello non nasce come sistema specializzato su un dominio ristretto, ma come modello *foundation* per il trattamento del parlato.

Nel caso della presente tesi, questo aspetto è particolarmente rilevante. Il dominio agricolo presenta condizioni difficilmente controllabili: parlato spontaneo, rumori ambientali, terminologia tecnica, nomi di prodotti e riferimenti contestuali noti all'operatore ma non sempre esplicitati. Addestrare un modello ASR specifico per questo dominio richiederebbe un dataset annotato, risorse computazionali e competenze specialistiche. L'uso di Whisper tramite API rappresenta quindi una scelta progettuale pragmatica: sfruttare un modello generalista robusto, accettando i limiti legati all'assenza di adattamento specifico.

2.2 Addestramento *weakly supervised* su larga scala

Una delle caratteristiche principali di Whisper è il tipo di addestramento utilizzato. Il modello è stato addestrato su circa 680.000 ore di audio associato a trascrizioni raccolte da fonti disponibili su Internet. Gli autori definiscono questo approccio come *large-scale weak supervision*, cioè supervisione debole su larga scala.

La supervisione è detta "debole" perché le trascrizioni utilizzate non sono necessariamente perfette o prodotte secondo standard rigorosi di annotazione. A differenza dei dataset accademici tradizionali, spesso più piccoli ma accuratamente validati, il dataset di Whisper privilegia la scala e la varietà. La scelta si basa sull'ipotesi che un modello addestrato su dati molto diversificati possa acquisire una maggiore robustezza rispetto a parlanti, ambienti acustici, lingue e domini differenti.

Questa impostazione comporta un compromesso. Da un lato, dati raccolti dal web possono contenere errori, trascrizioni incomplete, allineamenti imperfetti o testo generato automaticamente da altri sistemi ASR. Dall'altro, la grande quantità e varietà dei dati permette al modello di apprendere una distribuzione molto ampia del parlato reale. Il lavoro di Radford et al. mostra che questa scelta consente a Whisper di ottenere buone prestazioni in modalità *zero-shot*, cioè senza essere addestrato sullo specifico dataset di valutazione.

Per migliorare la qualità del dataset, gli autori applicano diverse procedure di filtraggio. Vengono rimossi, ad esempio, molti casi di trascrizioni generate automa-

ticamente, trascrizioni non coerenti con la lingua dell'audio e duplicati. Inoltre, l'audio viene suddiviso in segmenti di 30 secondi, ciascuno associato alla porzione di trascrizione corrispondente. Questa scelta semplifica il *training* e definisce anche una caratteristica importante del modello: Whisper lavora internamente su finestre audio di durata limitata.

Nel contesto della presente tesi, l'approccio *weakly supervised* è importante perché spiega sia i punti di forza sia i limiti del modello. La varietà dei dati di *training* contribuisce alla robustezza generale, utile in contesti non controllati come quello agricolo. Tuttavia, la natura generalista del dataset non garantisce che termini altamente specifici, come nomi commerciali di prodotti agricoli o lessico agronomico, siano sempre riconosciuti correttamente.

2.3 Architettura encoder-decoder Transformer

Whisper utilizza un'architettura *encoder–decoder* basata su Transformer (Figure 2.1). Questa scelta lo colloca nella famiglia dei modelli *sequence-to-sequence*, cioè modelli che trasformano una sequenza in input in una sequenza in output. Nel caso specifico, la sequenza in input è una rappresentazione dell'audio, mentre la sequenza in output è il testo trascritto.

Il segnale audio viene prima ricampionato a 16 kHz e trasformato in uno spettrogramma *log-Mel*. Questa rappresentazione, già introdotta nel Capitolo 1, descrive l'evoluzione delle componenti frequenziali del segnale nel tempo ed è particolarmente adatta all'elaborazione del parlato. Nel paper originale, lo spettrogramma utilizzato è composto da 80 canali Mel e viene calcolato su finestre temporali brevi.

L'encoder ha il compito di trasformare la rappresentazione audio in una serie di rappresentazioni interne. In altre parole, l'encoder estrae dal segnale audio informazioni utili per il riconoscimento del parlato. Il decoder, invece, genera il testo in modo autoregressivo: produce un token alla volta, condizionandosi sia sull'audio codificato dall'encoder sia sui token generati precedentemente.

Questa struttura è diversa da quella di alcuni sistemi ASR basati su CTC, in cui il modello produce direttamente una sequenza allineata al segnale audio. Nel caso di Whisper, il decoder funziona in modo simile a un modello linguistico condizionato dall'audio. Questo gli consente non solo di riconoscere i suoni, ma anche di produrre testo più naturale, con punteggiatura, maiuscole e una forma complessivamente più leggibile.

Per il testo, Whisper utilizza una tokenizzazione di tipo *byte-level BPE*. Nei modelli multilingue, il vocabolario viene adattato per evitare una frammentazione eccessiva

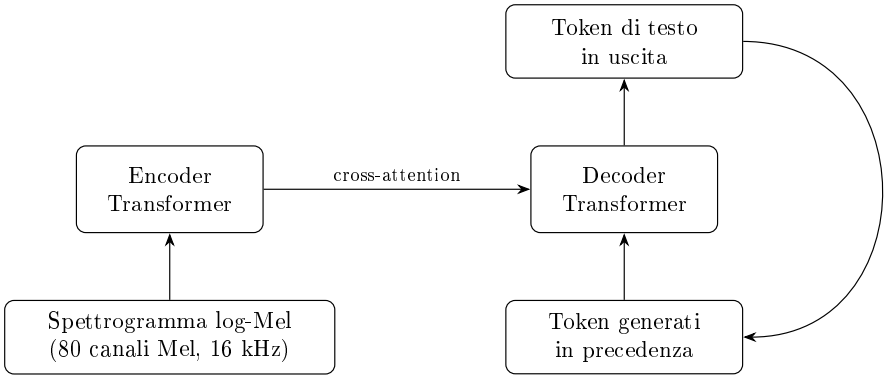

Figura 2.1: Architettura *encoder–decoder* di Whisper. L'encoder Transformer trasforma lo spettrogramma *log-Mel* in rappresentazioni interne; il decoder Transformer genera il testo un token alla volta in modo autoregressivo, condizionandosi sulle rappresentazioni dell'encoder tramite *cross-attention* e sui token prodotti in precedenza. Adattato da Radford et al. [1].

nelle lingue diverse dall'inglese. Questo aspetto è rilevante perché Whisper non è progettato soltanto per l'inglese, ma per un uso multilingue.

Gli autori addestrano una famiglia di modelli di dimensioni diverse: Tiny, Base, Small, Medium e Large. Le dimensioni variano da alcune decine di milioni fino a oltre un miliardo di parametri. In generale, modelli più grandi tendono a ottenere prestazioni migliori, soprattutto in compiti multilingue e di traduzione, anche se con un maggiore costo computazionale.

È importante notare che la dimensione del modello non è un parametro liberamente selezionabile quando Whisper viene utilizzato tramite l'API di OpenAI, come avviene in *Traccial'Abilità*. L'endpoint di trascrizione espone infatti Whisper attraverso un modello specifico, identificato dalla stringa `whisper-1`, che secondo OpenAI corrisponde al modello *large-v2* [2;3]. L'utilizzo via API non consente quindi di scegliere tra le diverse dimensioni della famiglia descritte sopra, né di accedere a varianti più recenti come *large-v3*: si ottiene il modello ospitato da OpenAI. Va inoltre osservato che la stessa API espone anche modelli di trascrizione più recenti basati su GPT-4o, ad esempio `gpt-4o-transcribe` e `gpt-4o-mini-transcribe`, che non sono però varianti di Whisper e non rientrano quindi nell'oggetto di questa analisi.

2.4 Formato multitask

Un elemento distintivo di Whisper è il suo formato *multitask*. Il modello non viene addestrato esclusivamente per trascrivere il parlato in una singola lingua, ma per gestire più compiti collegati al trattamento del segnale vocale.

Tra i compiti considerati rientrano:

- trascrizione nella stessa lingua del parlato;
- traduzione del parlato verso l'inglese;
- identificazione della lingua;
- rilevamento di segmenti senza parlato;
- generazione di timestamp;
- gestione di contesto testuale precedente.

Per permettere a un unico modello di gestire compiti diversi, Whisper utilizza token speciali inseriti nel decoder (Figure 2.2). Questi token specificano, ad esempio, la lingua, il tipo di operazione richiesta e la presenza o assenza di timestamp. In questo modo, la trascrizione diventa parte di un formato più generale, in cui il modello riceve indicazioni sul compito da svolgere e produce una sequenza coerente con tale compito.

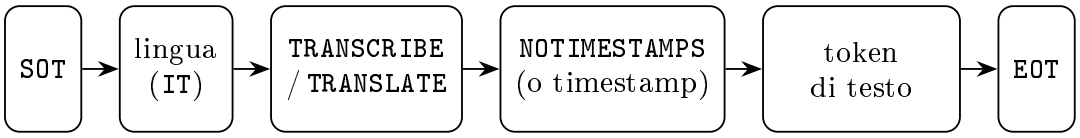

Figura 2.2: Formato *multitask* di Whisper. La sequenza di token in ingresso al decoder specifica, tramite token speciali, l'inizio della trascrizione (`SOT`), la lingua, il compito (`TRANSCRIBE` oppure `TRANSLATE`) e la presenza o assenza di timestamp (`NOTIMESTAMPS`), seguiti dai token di testo e dal token di fine (`EOT`); la sequenza può inoltre essere preceduta dai token del contesto testuale precedente. Adattato da Radford et al. [1].

Questa impostazione è significativa perché riduce la necessità di costruire pipeline separate per ogni componente. Nei sistemi ASR tradizionali, attività come riconoscimento della lingua, *voice activity detection*, trascrizione e allineamento temporale possono richiedere moduli distinti. Whisper, invece, integra molte di queste funzionalità in un unico modello *sequence-to-sequence*.

Per la presente tesi, il compito principale è la trascrizione automatica. Tuttavia, il carattere *multitask* del modello spiega perché Whisper sia adatto a essere integrato in applicazioni reali: non produce soltanto una sequenza grezza di parole, ma può restituire un testo più strutturato e leggibile. Questo è utile in *Traccial'Abilità*, dove la trascrizione deve essere consultabile e potenzialmente rielaborabile in funzione del Quaderno di Campagna dell'Agricoltore.

2.5 Zero-shot e robustezza

Uno dei risultati principali del paper di Radford et al. riguarda la capacità di Whisper di operare in modalità *zero-shot*. Nel lavoro originale, gli autori valutano il modello su numerosi dataset senza usare i dati di *training* degli stessi per adattarlo. L'obiettivo è misurare la generalizzazione del modello, non la sua capacità di ottimizzarsi su una distribuzione specifica.

Questa distinzione è importante. Un modello può ottenere prestazioni eccellenti su un benchmark dopo essere stato addestrato o *fine-tuned* su dati molto simili, ma fallire quando viene applicato a contesti reali diversi. Whisper, invece, viene valutato per la sua capacità di mantenere prestazioni buone su domini differenti.

Nel paper viene mostrato che Whisper, pur non essendo sempre il migliore su dataset specifici in condizioni controllate, tende a comportarsi meglio di molti modelli supervisionati quando viene applicato a dati fuori distribuzione. Gli autori evidenziano in particolare la differenza tra prestazioni *in-distribution* e *out-of-distribution*: i modelli ottimizzati su un singolo dataset possono risultare molto accurati su quel dataset, ma meno robusti in ambienti diversi.

Questa proprietà è centrale per il caso *Traccial'Abilità*. L'audio registrato da operatori agricoli non appartiene a un benchmark standard e non è prodotto in condizioni di laboratorio. Può contenere pause, rumori, esitazioni, parole tecniche e riferimenti contestuali. In assenza di un dataset agricolo annotato, l'uso *zero-shot* di Whisper rappresenta quindi una soluzione coerente con la filosofia del modello.

È però necessario evitare una conclusione eccessivamente forte. *Zero-shot* non significa assenza di errori. Significa piuttosto che il modello è progettato per fornire buone prestazioni generali senza adattamento specifico. Quando il dominio contiene lessico molto specialistico, come nel caso agricolo, rimangono possibili errori su termini rari o non sufficientemente rappresentati nei dati di addestramento.

2.6 Robustezza al rumore e condizioni non controllate

Un altro aspetto rilevante del paper originale riguarda la robustezza al rumore. Gli autori confrontano Whisper con altri modelli ASR valutando la degradazione delle prestazioni in presenza di rumore additivo, come rumore bianco o rumore ambientale simile a quello di un locale affollato. I risultati mostrano che Whisper tende a degradare più lentamente rispetto a modelli addestrati principalmente su dataset più controllati.

Questa osservazione è importante per applicazioni reali. In molti scenari pratici, l'audio non viene registrato in studio, ma tramite dispositivi mobili, microfoni non professionali e in presenza di interferenze. In agricoltura, rumori come vento, macchinari, movimento dell'operatore o suoni ambientali possono influire sulla qualità della registrazione.

La robustezza di Whisper al rumore costituisce quindi un vantaggio per *Traccia-l'Abilità*. Il sistema non può assumere che l'operatore registri sempre in condizioni acustiche ottimali. L'utilizzo di un modello addestrato su un'ampia varietà di condizioni audio rende più plausibile ottenere trascrizioni utili anche in scenari non perfettamente controllati.

Tuttavia, anche in questo caso è necessario interpretare correttamente il ruolo della trascrizione. Nel progetto analizzato, l'audio viene sempre conservato insieme al testo generato. Questo significa che, se il rumore produce un errore di trascrizione, l'informazione originaria non viene persa: può essere verificata riascoltando la registrazione. La robustezza del modello riduce quindi il carico di correzione, ma non elimina la necessità di mantenere la registrazione come fonte primaria.

2.7 Trascrizione di audio lunghi

Whisper è addestrato su segmenti audio di 30 secondi. Questa scelta è compatibile con molti dataset accademici, composti da frasi o brevi *utterance*, ma introduce una sfida quando si devono trascrivere registrazioni più lunghe. Nel paper originale, gli autori affrontano questo problema proponendo strategie di trascrizione *long-form* basate sulla suddivisione dell'audio in finestre successive e sull'uso dei timestamp prodotti dal modello.

La trascrizione di audio lunghi può presentare alcuni problemi specifici. Un errore in una finestra può influenzare la finestra successiva, soprattutto quando il modello utilizza il testo precedente come contesto. Inoltre, possono verificarsi ripetizioni, omissioni di parole all'inizio o alla fine dei segmenti, oppure problemi di allineamento temporale.

Per mitigare questi problemi, il paper descrive alcune euristiche di *decoding*, tra cui *beam search*, *temperature fallback*, uso della probabilità associata al token di assenza di parlato e condizionamento sul testo precedente. Queste tecniche sono pensate per rendere più stabile la trascrizione di registrazioni estese.

Nel caso di *Traccial'Abilità*, questo punto è rilevante ma va contestualizzato. Le registrazioni degli operatori agricoli sono finalizzate alla descrizione di attività operative e non alla trascrizione di conversazioni lunghe o riunioni. Tuttavia, anche note

vocali di alcuni minuti possono rientrare nel problema della trascrizione *long-form*. La scelta di salvare sempre l'audio originale permette di gestire eventuali errori di segmentazione o di allineamento, mantenendo la possibilità di verifica.

2.8 Limiti del modello

Il paper di Radford et al. non presenta Whisper come un sistema privo di limiti. Al contrario, gli autori discutono esplicitamente alcune criticità. Tra queste, una delle più importanti riguarda le allucinazioni (*hallucinations*): il modello può generare testo plausibile ma non effettivamente presente nell'audio. Questo fenomeno è particolarmente rilevante nei modelli *sequence-to-sequence*, in cui il decoder ha caratteristiche simili a un modello linguistico e può quindi produrre sequenze fluenti anche quando il segnale audio è ambiguo o degradato.

Altri limiti riguardano la trascrizione di audio lunghi, la possibilità di ripetizioni, omissioni di parole e difficoltà nella gestione di lingue o domini meno rappresentati nel dataset di addestramento. Gli autori osservano inoltre che le prestazioni sulle diverse lingue dipendono fortemente dalla quantità di dati disponibili per ciascuna lingua nel dataset di *training*. Le lingue con meno dati tendono ad avere prestazioni inferiori.

Per la presente tesi, il limite più rilevante riguarda il lessico di dominio. Whisper è robusto sul parlato generale, ma non è necessariamente ottimizzato per riconoscere termini specialistici dell'agricoltura. Nomi commerciali di prodotti, sigle, varietà colturali o trattamenti possono essere confusi con parole più comuni o trascritti in modo errato. Questo tema sarà ripreso nel Capitolo 3, dove vengono analizzati i lavori successivi su *contextual biasing*, *rare-word recognition*, *prompting* e adattamento di dominio.

Va anticipato che l'API di OpenAI, pur non consentendo *fine-tuning*, mette a disposizione un parametro testuale, `prompt`, che permette di orientare la trascrizione verso un determinato vocabolario o stile. Come documentato da OpenAI[2], per il modello `whisper-1` vengono considerati soltanto gli ultimi 224 token del *prompt*, che va inteso come un suggerimento e non come un vincolo rigido. Questo parametro rappresenta tuttavia il principale meccanismo di adattamento leggero effettivamente disponibile in uno scenario basato su API ed è alla base delle tecniche di *prompting* e di iniezione di terminologia discusse nel Capitolo 3.

Nel caso di *Traccial'Abilità*, questi limiti non invalidano l'uso del modello, ma ne definiscono il ruolo corretto. La trascrizione automatica deve essere considerata come un supporto all'elaborazione e non come sostituto dell'audio. La conservazione

della registrazione consente di verificare e correggere il testo quando necessario, mantenendo la tracciabilità dell'origine dell'informazione.

2.9 Rilevanza di Whisper per Traccial'Abilità

Alla luce delle caratteristiche descritte, Whisper risulta adatto al caso *Traccial'Abilità* per tre motivi principali.

In primo luogo, il modello può essere utilizzato senza addestramento specifico. Questo è fondamentale in un contesto in cui non è disponibile un corpus agricolo annotato con audio e trascrizioni validate. L'integrazione tramite API consente di introdurre rapidamente una funzionalità avanzata di trascrizione automatica senza dover sviluppare un'infrastruttura di *machine learning* dedicata.

In secondo luogo, Whisper è progettato per essere robusto rispetto a condizioni audio e domini diversi. Questo lo rende più adatto rispetto a modelli fortemente ottimizzati su dataset specifici ma meno capaci di generalizzare. Nel contesto agricolo, dove l'audio può essere registrato in condizioni variabili, questa robustezza è un requisito pratico importante.

In terzo luogo, il modello produce trascrizioni generalmente leggibili e adatte a un uso operativo. Nel progetto considerato, il testo trascritto può facilitare la consultazione delle registrazioni e supportare l'elaborazione delle informazioni richieste per il Quaderno di Campagna dell'Agricoltore. La trascrizione riduce quindi lo sforzo necessario per trasformare una nota vocale in informazione strutturabile.

Il punto critico rimane l'adattamento al dominio. Whisper è una base solida, ma l'ambito agricolo richiede attenzione al riconoscimento di termini specifici. Per questo motivo, l'uso del modello nel progetto deve essere interpretato come un compromesso tra efficacia, semplicità di integrazione e limiti del riconoscimento automatico. La letteratura successiva a Radford, analizzata nel Capitolo 3, si concentra proprio su questi aspetti: migliorare la capacità di Whisper di gestire lessico specialistico, termini rari e contesto applicativo senza necessariamente ricorrere a un *fine-tuning* completo.

In sintesi, Whisper rappresenta una soluzione coerente con gli obiettivi di *Traccial'Abilità*: semplificare il lavoro di raccolta ed elaborazione delle informazioni agricole, mantenendo l'audio come fonte primaria e usando la trascrizione come livello aggiuntivo di accessibilità, consultazione e supporto operativo.

Capitolo 3

Stato dell'arte

Il lavoro di Radford et al.[1] ha reso disponibile un modello di riconoscimento del parlato robusto e generalista, ma ha al tempo stesso reso evidente un limite strutturale: un modello addestrato per funzionare *out of the box* su un'ampia varietà di domini non è ottimizzato per nessuno di essi in particolare e, di fronte a lessico specialistico, nomi propri o termini rari, può produrre errori sistematici. La letteratura successiva ha esplorato in modo estensivo questo limite, proponendo tecniche per migliorare le prestazioni di Whisper su domini specifici e analizzandone in maniera critica i modi di fallimento.

Per organizzare questi contributi non è sufficiente distinguerli in base al problema affrontato (lessico, rumore, lingua); è altrettanto importante distinguerli in base a *quanto accesso al modello richiedono*. Alcune tecniche presuppongono la possibilità di ri-addestrare i pesi; altre intervengono sul processo di decodifica o sugli stati interni del modello; altre ancora agiscono unicamente sull'input testuale fornito al modello. Questa distinzione è decisiva in quanto il caso di studio analizzato utilizza Whisper tramite l'API ospitata di OpenAI, che non espone né i pesi né le probabilità dei token e mette a disposizione, come unico strumento di condizionamento, un breve prompt testuale (al massimo 224 token per `whisper-1`, come discusso nel Capitolo 2). Le sezioni seguenti ripercorrono la letteratura raggruppandola per famiglia di tecniche, evidenziando per ciascuna il livello di accesso richiesto.

3.1 Il problema dell'adattamento di dominio

Il problema comune a gran parte dei lavori considerati è il riconoscimento di parole rare e di terminologia di dominio. I sistemi ASR *end-to-end*, pur raggiungendo prestazioni elevate sul parlato comune, tendono a confondere termini poco frequenti

con parole più comuni foneticamente simili. Gli esempi riportati in letteratura sono illuminanti: un sistema basato su Whisper può trascrivere *morel* (un fungo) come *moral*, parola assai più frequente e quasi omofona[10], oppure *inpatient* come *impatient* in ambito medico[6]. La radice del problema è che tali termini sono sparsi o assenti nei dati di addestramento, e il decoder di Whisper, che si comporta come un modello linguistico condizionato dall'audio, tende a privilegiare le sequenze più probabili a priori.

Questo fenomeno è particolarmente critico nei domini verticali, dove proprio i termini rari — nomi commerciali, sigle, entità nominate — sono quelli che veicolano l'informazione più rilevante. Un errore su una parola comune è spesso irrilevante, mentre un errore su un nome di prodotto o su un principio attivo può compromettere l'utilità della trascrizione. È questa asimmetria a motivare sia le tecniche di adattamento descritte di seguito, sia la proposta di metriche di valutazione che pesino diversamente i termini di dominio, discusse nella Sezione 3.6.

3.2 Contextual biasing e iniezione di terminologia

Una prima e ampia famiglia di tecniche è quella del *contextual biasing*: orientare l'output del modello verso un vocabolario atteso, fornito sotto forma di lista di parole o frasi rilevanti. Un approccio classico, precedente a Whisper ma richiamato in molti di questi lavori, è la *shallow fusion*, che costruisce un trasduttore a stati finiti a partire da una lista di parole chiave e ne incrementa i punteggi durante la ricerca in fase di decodifica[5]. Tale schema richiede però l'accesso al processo di decodifica del modello.

Diversi lavori adattano questa idea a Whisper. *Improving Speech Recognition with Jargon Injection*[4] rappresenta il gergo di dominio in una struttura ad albero (*trie*) e forza la decodifica del modello a concentrarsi su tali termini modificando la probabilità dei token generati, combinando questa manipolazione con l'uso del gergo come prompt; gli esperimenti su dati giapponesi e inglesi mostrano un miglioramento, in particolare sui dati di dominio. *CB-Whisper*[5] introduce invece un modulo di *keyword spotting* a vocabolario aperto che, sfruttando rappresentazioni dell'encoder di Whisper e tecniche di sintesi vocale, individua le entità definite dall'utente prima della decodifica e le inietta nel decoder tramite prompt costruiti con suggerimenti sulla forma parlata; il metodo migliora in modo sostanziale il *mixed-error-rate* e il richiamo delle entità su dataset in inglese, cinese e in scenari di *code-switching*. *Keyword-Guided Adaptation*[7] adotta anch'esso un modello di keyword spotting che genera dinamicamente prompt per guidare il decoder, con due varianti — una che effettua il fine-tuning del decoder e una che apprende un prefisso di prompt — e riporta un miglioramento medio complessivo del WER del 5,1% rispetto a Whi-

sper, con una capacità di adattamento che generalizza anche a lingue non viste in addestramento.

Altri lavori puntano esplicitamente a evitare la modifica dei pesi. *Speech-enriched Memory* [6] propone un algoritmo applicabile in fase di inferenza che indicizza una lista di entità rare in una memoria, sintetizzando l'audio di ciascuna voce e memorizzando gli stati acustici e linguistici interni del modello; durante la trascrizione una ricerca per vicini più prossimi, vincolata a corrispondenze a livello di parola per evitare falsi positivi, migliora significativamente il riconoscimento dei termini rari. *Contextual Biasing without Explicit Fine-Tuning* [8] guida l'output verso un vocabolario specifico integrando una struttura ad albero di prefissi di tipo neuro-simbolico, addestrata su un piccolo dataset ma senza alterare i parametri del modello, ottenendo una riduzione del WER in ambito marittimo. Infine, *Improving Rare-Word Recognition of Whisper in Zero-Shot Settings* [20] adotta una strategia supervisionata: effettua il fine-tuning di Whisper su un'istruzione di contextual biasing usando 670 ore di Common Voice inglese, ottenendo un miglioramento del 45,6% nel riconoscimento di parole rare e del 60,8% per parole non viste durante il fine-tuning, con una capacità di biasing che generalizza persino a lingue non incontrate in addestramento.

Pur affrontando lo stesso problema, questi metodi si collocano a livelli di accesso molto diversi: l'iniezione di gergo agisce sulle probabilità dei token, CB-Whisper e la memoria arricchita richiedono le rappresentazioni dell'encoder o gli stati interni, l'adattamento guidato da keyword e il riconoscimento di parole rare richiedono l'addestramento. Nessuna di queste forme di intervento è disponibile attraverso l'API ospitata, che non espone le probabilità dei token, gli stati interni o i pesi del modello. L'unico elemento di queste tecniche riconducibile allo scenario basato su API è l'uso dei termini come testo nel prompt — un ingrediente che alcuni di questi metodi adottano, ma sempre in combinazione con meccanismi più profondi.

3.3 Prompting e in-context learning

Una seconda famiglia di lavori si concentra su tecniche che evitano la modifica dei pesi agendo, in modi diversi, sul contesto fornito al modello. È però necessario distinguere con attenzione tecniche che il linguaggio comune accomuna sotto il termine «prompting», ma che presuppongono requisiti molto differenti.

Can Whisper Perform Speech-Based In-Context Learning? [9] studia la capacità di Whisper di adattarsi a partire da esempi forniti al momento dell'inferenza, senza discesa del gradiente. L'approccio proposto, basato su un piccolo numero di esempi di parlato etichettati e su una selezione degli esempi per vicini più prossimi, riduce il WER in misura considerevole su dialetti cinesi — in media del 32,3%, e del 36,4%

con la selezione degli esempi. Si tratta tuttavia di un condizionamento basato su *esempi di parlato*, ossia su coppie audio-trascrizione fornite come contesto, modalità non supportata dall'endpoint di trascrizione dell'API, che accetta un singolo file audio e un prompt testuale.

Improving Domain-Specific ASR with LLM-Generated Contextual Descriptions[10] fornisce al modello descrizioni testuali del contesto, generate automaticamente da un modello linguistico a partire da semplici metadati quando non sono disponibili descrizioni redatte manualmente; gli esperimenti mostrano che le descrizioni generate da LLM risultano più efficaci di quelle scritte da esseri umani. Il metodo completo include anche tecniche di addestramento aggiuntive (fine-tuning del decoder e perturbazione del contesto), ma il nucleo dell'idea — fornire una descrizione testuale come contesto — è compatibile, almeno in linea di principio, con il prompt testuale dell'API. Va chiarito che *Exploring the Potential of Prompting Methods in Low-Resource Speech Recognition with Whisper*[23] non riguarda il prompting testuale ma il *prompt tuning*: l'apprendimento, tramite addestramento, di prompt continui (vettori) che fungono da parametri aggiuntivi. Il lavoro mostra che tali prompt appresi raggiungono prestazioni competitive con il fine-tuning completo e con LoRA usando un numero ridotto di parametri addestrabili, ma resta una tecnica che richiede l'addestramento e l'accesso al modello, non assimilabile all'inserimento di testo nel prompt.

La distinzione è cruciale: «prompting» può indicare sia l'apprendimento di prompt continui (una forma di *fine-tuning* efficiente, non disponibile via API), sia l'inserimento di testo nel parametro `prompt` (disponibile via API, ma limitato a un breve suggerimento). Solo la seconda accezione è praticabile nello scenario considerato, ed è in ogni caso un meccanismo dichiaratamente debole, inteso come suggerimento e non come vincolo.

3.4 Fine-tuning e adattamento supervisionato

Una terza famiglia di lavori accetta esplicitamente la modifica dei pesi e ne studia l'efficacia. *Exploration of Whisper Fine-tuning Strategies for Low-Resource ASR*[22] confronta sistematicamente cinque strategie di fine-tuning su sette lingue a basse risorse, mostrando che tutte migliorano significativamente le prestazioni di Whisper, ma che la loro idoneità ed efficacia pratica variano, richiedendo una scelta attenta in funzione del caso d'uso e delle risorse disponibili. *Fine-tuning Whisper on Low-Resource Languages for Real-World Applications*[18], prendendo il tedesco svizzero come caso di studio, introduce un metodo per generare un corpus *long-form* a partire da dati a livello di frase, preservando la capacità del modello di gestire audio lunghi

e di produrre segmentazione tramite timestamp; il modello risultante stabilisce un nuovo stato dell'arte per il tedesco svizzero.

Due lavori sono particolarmente interessanti poiché riducono progressivamente i requisiti di dati, pur mantenendo quello dell'accesso ai pesi. *Domain-Specific Adaptation through Text-Only Fine-Tuning*[19] effettua il fine-tuning del solo decoder utilizzando esclusivamente testo di dominio, tramite *embedding* di bias addestrabili nei meccanismi di cross-attention estesi con un instradamento a *mixture-of-experts*, ottenendo un miglioramento relativo del WER fino al 56% senza alcun dato audio. *A Domain Adaptation Framework with Only Synthetic Data*[21] elimina del tutto la necessità di dati reali: un modello linguistico genera testi di dominio, che vengono convertiti in parlato tramite sintesi vocale e usati per il fine-tuning di Whisper con adattatori LoRA, con una strategia di decodifica che fonde in un unico passaggio le predizioni di più adattatori; il metodo riduce il WER del 10–17% sui domini bersaglio, con una regressione minima (circa l'1%) fuori dominio.

Questa famiglia rappresenta lo strumento più efficace per un adattamento profondo al dominio, e mostra una traiettoria che allenta i requisiti sui dati — da audio etichettato, a solo testo, a dati interamente sintetici. Tuttavia il requisito che non viene mai meno è l'accesso ai pesi del modello e alle risorse computazionali per il loro aggiornamento: nessuna di queste tecniche è applicabile a un modello utilizzato esclusivamente attraverso un'API di inferenza ospitata.

3.5 Robustezza, allucinazioni e modi di fallimento

Un quarto filone analizza in modo critico i limiti del modello, indipendentemente dalle tecniche di adattamento. Il contributo più rilevante per questa tesi è *Careless Whisper: Speech-to-Text Hallucination Harms*[12], che valuta Whisper su trascrizioni reali e rileva che circa l'1% delle trascrizioni contiene intere frasi allucinate, ossia non presenti in alcuna forma nell'audio sottostante. L'analisi tematica mostra che il 38% di tali allucinazioni include contenuti esplicitamente dannosi. Particolarmente significativo, in relazione al caso di studio, è il fatto che le allucinazioni si presentino in misura sproporzionata in corrispondenza di porzioni audio con lunghe durate non vocali, cioè in presenza di silenzio: è esattamente la condizione in cui il modello tende a generare formule spurie, e che il sistema descritto nel Capitolo 4 intercetta con uno specifico filtro.

Altri lavori caratterizzano la degradazione delle prestazioni in condizioni non ideali. *Benchmarking Whisper Under Diverse Audio Transformations and Real-Time Constraints*[13] valuta Whisper sotto rumore bianco e gaussiano, riverbero, *time stretch* e *pitch shift*, oltre che al variare della lunghezza dei segmenti, e rileva che il mo-

dello, pur gestendo bene il rumore lieve e l'audio pulito, degrada rapidamente sotto trasformazioni più severe, in particolare quando si utilizzano segmenti più brevi. Sul versante delle capacità native del modello, *Turning Whisper into Real-Time Transcription System*[14] osserva che Whisper non è progettato per la trascrizione in tempo reale e propone un'implementazione (*Whisper-Streaming*) basata su una politica di accordo locale, raggiungendo una latenza di 3,3 secondi su parlato lungo non segmentato; *Whisper-TAD*[15] estende invece il modello, tramite fine-tuning, perché svolga in un unico sistema trascrizione, allineamento e diarizzazione. Questi ultimi due lavori riguardano funzionalità (streaming, diarizzazione) non centrali per il caso di studio, ma documentano i confini delle capacità native di Whisper.

3.6 Valutazioni linguistiche e in contesto agricolo

Un ultimo gruppo di lavori valuta Whisper in contesti linguistici e applicativi specifici, e fornisce i riferimenti più direttamente pertinenti al caso di studio. *Does Whisper Understand Swiss German?*[16] valuta il modello su un dialetto presumibilmente assente dai dati di addestramento, attraverso metriche automatiche (WER e BLEU), analisi qualitativa e una valutazione umana con 28 partecipanti. Il risultato è in larga parte positivo: Whisper si rivela un sistema valido per il tedesco svizzero, a condizione che l'output desiderato sia in tedesco standard. Questo dettaglio è importante e va interpretato con attenzione: il modello non trascrive il dialetto in modo letterale ma lo *normalizza*, producendo una traduzione in tedesco standard. Più che un limite linguistico, lo studio evidenzia dunque una notevole robustezza zero-shot a varietà non viste, con la conseguenza che l'output può discostarsi dalla forma esatta del parlato.

A Comparative Study of Speech-to-Text for Italian[17] è di interesse diretto perché riguarda la lingua del caso di studio. Lo studio confronta sull'italiano (dataset Common Voice) i modelli Whisper Tiny e Whisper Base con il sistema Vosk, affiancati da metodi di correzione ortografica, in vista di applicazioni su dispositivi con risorse limitate. I risultati indicano che, dopo un fine-tuning sul dataset italiano, Whisper Base supera sia Whisper Tiny sia Vosk; va però osservato che in assenza di adattamento (zero-shot) i modelli Whisper di piccole dimensioni qui testati risultano più deboli — con un WER del 45% per Whisper Base, comparabile a quello di Vosk — e che il guadagno deriva proprio dal fine-tuning. Lo studio mostra inoltre che in ambienti poco rumorosi l'impatto della correzione ortografica è minimo, a indicare una buona robustezza del modello anche senza post-elaborazione. Due osservazioni sono rilevanti per questa tesi. In primo luogo, lo studio impiega varianti di piccole dimensioni (Tiny e Base, rispettivamente 39 e 74 milioni di parametri), mentre l'API ospitata espone large-v2, di ordini di grandezza più capiente: la competitività

dell'italiano in modalità zero-shot va quindi attesa, per il caso di studio, dal modello grande e non dai modelli piccoli qui valutati. In secondo luogo, anche in condizioni controllate emerge che la dimensione del modello incide in misura determinante sulle prestazioni.

Il riferimento più calzante è però *Benchmarking Automatic Speech Recognition for Indian Languages in Agricultural Contexts* [11], che valuta sistemi ASR — tra cui proprio Whisper nella versione esposta dall'API di OpenAI — su registrazioni agricole reali in hindi, telugu e odia. Lo studio introduce una metrica pensata per il dominio, l'*Agriculture Weighted Word Error Rate* (AWWER), che pesa diversamente gli errori sui termini agricoli rispetto al vocabolario generale, proprio in virtù dell'asimmetria discussa nella Sezione 3.1, e organizza gli errori in dodici categorie di dominio (tra cui nomi di colture, malattie e parassiti, nomi di prodotti chimici, varietà colturali e unità di misura). Su oltre diecimila registrazioni, Whisper via API si colloca nelle posizioni di coda: sul telugu l'output di Whisper via API risulta incoerente e non viene quindi riportato; per l'hindi ottiene un WER del 46,8% (settimo su dieci modelli, a fronte del 16,2% del miglior sistema), e per l'odia, lingua a più basse risorse, un WER del 125,6%, valore superiore al 100% dovuto all'elevato numero di inserzioni. Lo studio mostra inoltre che la selezione del miglior interlocutore tramite diarizzazione riduce sensibilmente il WER sulle registrazioni con più parlanti.

Questi risultati vanno letti con la dovuta cautela, poiché riguardano lingue indiane e non l'italiano, che è una lingua a risorse più elevate per la quale ci si attendono prestazioni migliori; sarebbe quindi scorretto trasferire i valori assoluti al caso di studio. Resta però solido il dato qualitativo, e per giunta ottenuto sul medesimo strumento utilizzato in questa tesi: un modello generalista impiegato in modalità *zero-shot* via API è fragile proprio sul lessico di dominio agricolo, e la sua qualità degrada marcatamente al peggiorare delle condizioni acustiche e della copertura linguistica.

3.7 Sintesi: applicabilità al caso di studio

La rassegna mostra un panorama di tecniche efficaci ma disposte lungo uno spettro di accesso al modello. All'estremo più potente si collocano il *fine-tuning* — completo, del solo decoder o tramite adattatori a basso rango — e le tecniche di *contextual biasing* che intervengono sulla decodifica o sugli stati interni: sono gli strumenti che producono i miglioramenti più consistenti sul lessico di dominio, ma richiedono l'accesso ai pesi o al processo di decodifica. In una posizione intermedia si trovano metodi che evitano la modifica dei pesi ma presuppongono comunque forme di accesso non banali, come la memoria arricchita dal parlato o l'apprendimento di

prompt continui. All'estremo più vincolato si colloca l'unico meccanismo realmente disponibile attraverso un'API ospitata: l'inserimento di testo — termini di dominio o descrizioni del contesto — in un breve prompt.

Il caso di studio descritto nel Capitolo 4 si colloca precisamente a questo estremo. La letteratura fornisce così le coordinate per le due domande affrontate nel Capitolo 5: quali fra le tecniche qui esaminate siano realisticamente trasferibili — eventualmente in una forma adattata e più debole — a uno scenario basato esclusivamente sull'API, e quanto la scelta progettuale di conservare l'audio come fonte primaria sia adeguata a fronte dei limiti, in particolare delle allucinazioni, documentati in questa rassegna.

Capitolo 4

Caso di studio: Traccial'Abilità

Il sistema *Traccial'Abilità* è stato sviluppato per supportare la raccolta e la consultazione delle informazioni operative relative alle attività agricole, con particolare riferimento alla successiva elaborazione del Quaderno di Campagna dell'Agricoltore (QDCA). Il QDCA richiede di documentare in modo dettagliato le operazioni effettuate sulle colture, i prodotti utilizzati e le informazioni necessarie alla tracciabilità delle attività aziendali. In questo contesto, il sistema non sostituisce il ruolo dell'agricoltore nella dichiarazione dell'attività, ma semplifica il processo con cui l'informazione viene acquisita, conservata, verificata ed elaborata.

Il presente capitolo descrive l'architettura del sistema, il modello dei dati, il flusso di acquisizione e trascrizione dell'audio, la gestione della trascrizione come strumento di supporto e il livello di collaborazione tra gli attori della filiera. La trattazione si concentra sul sottosistema di trascrizione automatica, che costituisce l'oggetto della tesi, lasciando sullo sfondo le ulteriori funzionalità della piattaforma non direttamente attinenti all'uso di Whisper.

4.1 Obiettivi e ruolo della trascrizione

L'operatore agricolo può registrare vocalmente una descrizione dell'attività svolta direttamente sul campo, nel momento in cui essa viene effettuata o in prossimità di esso. Il sistema conserva la registrazione e ne genera automaticamente una trascrizione testuale tramite il modello Whisper. Le trascrizioni così ottenute costituiscono un registro delle attività di campo, indicato come brogliaccio aziendale.

Il contributo del brogliaccio al QDCA è di tipo indiretto. Le informazioni in esso contenute non vengono riversate automaticamente nel Quaderno di Campagna: è l'operatore preposto alla compilazione che le consulta e le trasferisce manualmente

nel documento. La trascrizione svolge quindi una funzione di supporto, riducendo lo sforzo necessario per reperire e ricostruire le informazioni sulle attività svolte, ma non si sostituisce all'attività di compilazione né alla responsabilità di chi la esegue.

Un principio progettuale centrale è che la *source of truth* rimane sempre la registrazione audio. La trascrizione viene salvata insieme all'audio da cui deriva e ne è subordinata: in caso di dubbio, ambiguità o errore del modello, è sempre possibile risalire alla registrazione originale prodotta dall'operatore. Per rafforzare questa garanzia, di ogni file audio viene calcolato e memorizzato un hash crittografico (SHA-256), che ne consente la verifica di integrità nel tempo.

4.2 Architettura del sistema

Il sistema adotta un'architettura client–server in cui la logica applicativa risiede interamente nel backend e i client si limitano all'acquisizione e alla presentazione dei dati. I componenti principali sono:

- due client di acquisizione che condividono lo stesso backend: un'applicazione mobile sviluppata in Flutter, pensata per l'uso in campo, e una dashboard web per la consultazione e la gestione;
- un backend realizzato come plugin WordPress, che espone le funzionalità tramite un'API REST dedicata e gestisce autenticazione, autorizzazioni e persistenza;
- un sistema di archiviazione esterno basato su Dropbox, in cui vengono conservati i file audio ed eventuali allegati;
- l'integrazione con l'API di OpenAI per la trascrizione automatica.

La scelta di realizzare il backend come plugin WordPress risponde a un criterio di coerenza con l'infrastruttura già in uso in azienda: WordPress costituiva la piattaforma di gestione dei contenuti e dei siti aziendali già adottata e presidiata internamente. Sviluppare il sistema come estensione di tale piattaforma ha consentito di riutilizzare un ambiente di hosting, un modello di autenticazione e competenze di manutenzione già consolidati, riducendo i costi di integrazione e di gestione operativa rispetto all'introduzione di uno stack tecnologico nuovo. Questa decisione è coerente con l'impostazione progettuale complessiva del sistema, orientata a privilegiare soluzioni pragmatiche e a basso attrito di adozione — la stessa logica che, sul piano della trascrizione, motiva l'uso di Whisper tramite API anziché lo sviluppo di un modello dedicato.

L'archiviazione dei file su Dropbox risponde allo stesso criterio adottato per il backend: Dropbox era già il servizio di storage in uso in azienda — che ne è peraltro partner, con possibilità di rivendita delle licenze — il che ha consentito di appoggiarsi a un'infrastruttura di archiviazione e a un rapporto di licensing già consolidati, senza introdurre un nuovo sistema da integrare e mantenere. Coerentemente con questa scelta, i file audio non vengono memorizzati nel database del backend: in esso sono conservati soltanto i metadati dell'interazione, il percorso del file su Dropbox e l'hash SHA-256 della registrazione. I file sono organizzati su Dropbox secondo una struttura per utente e azienda e vengono resi disponibili ai client tramite link temporanei. L'unità informativa centrale del sistema, l'*interazione*, è descritta in dettaglio nella sezione seguente.

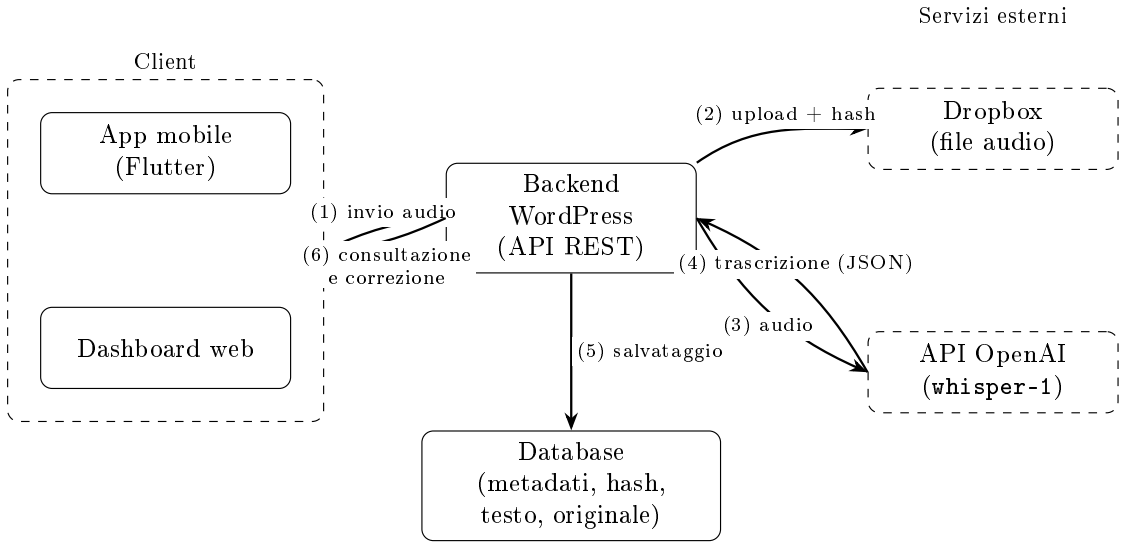

Figura 4.1: Architettura di *Traccial'Abilità* e flusso di acquisizione e trascrizione. Le componenti interne (backend, database) hanno bordo continuo, i servizi esterni (Dropbox, API OpenAI) bordo tratteggiato; gli archi numerati indicano l'ordine delle operazioni.

4.3 Modello dei dati e ciclo di vita dell'interazione

L'unità informativa centrale del sistema è l'*interazione*: ogni annotazione prodotta da un operatore, vocale o testuale, è rappresentata da un singolo record. Il campo `tipo` distingue le note vocali, che danno luogo a una trascrizione (`trascrizione`), dalle note inserite direttamente come testo (`testo`). Indipendentemente dal tipo, i dati associati a un'interazione si possono raggruppare in tre famiglie:

- **identità e relazioni**: l'azienda di riferimento, l'autore della nota, l'eventuale destinatario e, nel caso di una risposta, il riferimento all'interazione a cui essa si collega;
- **contenuto**: il testo dell'interazione e, conservata separatamente, la trascrizione originale prodotta dal modello;
- **riferimento alla fonte**: per le note vocali, il percorso del file audio su Dropbox, il tipo MIME, la durata, la dimensione e l'hash SHA-256 della registrazione.

A queste informazioni si aggiungono i due campi che ne descrivono lo stato, discussi più avanti. Coerentemente con quanto già osservato, i byte dell'audio non risiedono nel database: vi sono conservati soltanto i metadati e il riferimento al file. A un'interazione possono inoltre essere associati degli allegati: immagini o documenti PDF, fino a un massimo di dieci per interazione e di 20 MB ciascuno, archiviati su Dropbox con le stesse modalità dell'audio.

Lo stato di un'interazione è descritto su due livelli ortogonali. Il primo è lo **stato di lavorazione** (campo `stato`), che riflette il punto del processo di trascrizione in cui l'interazione si trova. Le note puramente testuali assumono lo stato *normale*. Per le note vocali, l'audio viene anzitutto caricato e l'interazione assume lo stato *caricato*; su di esso viene quindi tentata la trascrizione. Se il modello restituisce un testo, l'interazione passa allo stato *trascritto*; se la trascrizione fallisce o viene annullata dal filtro descritto nella Sezione 4.6, passa allo stato *errore*; se nessuna trascrizione viene prodotta, resta nello stato *caricato*. Una successiva correzione manuale del testo porta infine l'interazione nello stato *corretto*, reversibile ripristinando la trascrizione originale. Questo ciclo è illustrato nella Figure 4.2.

Il secondo livello è il **ciclo di vita logico** (campo `interaction_status`), che può assumere i valori *attiva*, *modificata* ed *eliminata* e governa la storia del record nel tempo. L'aspetto progettualmente più rilevante è che le modifiche non sono distruttive. Una correzione non sovrascrive l'interazione esistente: viene creata una nuova versione, marcata come *attiva*, mentre la versione di partenza viene marcata come *modificata* e resta conservata. Analogamente, l'eliminazione è puramente logica: l'interazione passa allo stato *eliminata* senza essere rimossa fisicamente. La trascrizione originale prodotta dal modello viene mantenuta in un campo dedicato e può essere ripristinata in qualsiasi momento; l'operazione di ripristino reintroduce il testo originale come nuova versione attiva, riportando lo stato di lavorazione a *trascritto*.

Questa organizzazione dei dati rende la provenienza dell'informazione verificabile dall'inizio alla fine. La registrazione audio resta la fonte primaria e la sua integrità

è garantita dall'hash crittografico; l'output grezzo del modello non viene mai perso; ogni correzione è tracciata e reversibile; nessun dato viene cancellato fisicamente. La trascrizione si configura così come un livello di supporto rivedibile, sovrapposto a una base documentale immutabile, coerentemente con il principio secondo cui essa assiste l'elaborazione ma non sostituisce mai la fonte.

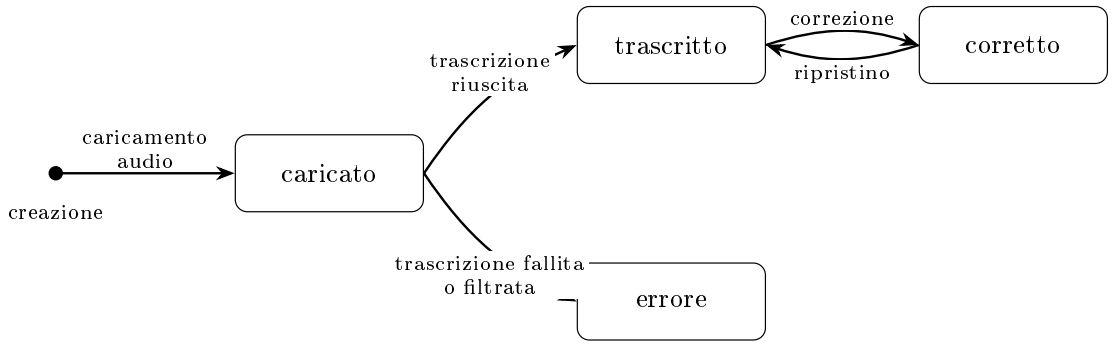

Figura 4.2: Ciclo di lavorazione (campo `stato`) di un'interazione audio. L'audio viene prima salvato (*caricato*); la trascrizione, tentata successivamente, porta allo stato *trascritto* se riesce o *errore* se fallisce o viene filtrata. Una correzione manuale porta allo stato *corretto*, reversibile tramite ripristino della trascrizione originale. Le note testuali assumono lo stato *normale*.

4.4 Acquisizione dell'audio e gestione della fonte

Il flusso di acquisizione, riassunto nella Figure 4.1, si articola nei seguenti passaggi: l'utente registra l'audio tramite l'app mobile o la dashboard web; il client invia il file al backend tramite l'API REST; il backend carica la registrazione su Dropbox e ne calcola l'hash; viene quindi tentata la trascrizione automatica; infine, il testo risultante viene salvato insieme al riferimento all'audio e reso disponibile per la consultazione e l'eventuale correzione.

Dato che l'app è pensata per l'uso sul campo, dove la connettività può non essere garantita, l'acquisizione è progettata in ottica *offline-first*: il client può registrare una nota anche in assenza di rete e trasmetterla in un momento successivo. Questo introduce un problema di datazione, poiché l'istante in cui la nota raggiunge il backend non coincide con quello in cui è stata effettivamente prodotta, e poiché l'orologio del dispositivo può essere impreciso. Per ricostruire un istante di creazione affidabile, il client trasmette due marche temporali, entrambe rilevate con il proprio orologio: il momento di creazione della nota t_c e il momento del suo invio t_i. Detto t_s l'istante di ricezione misurato dal server in tempo universale (UTC), che costituisce il riferimento autoritativo, il backend calcola l'istante di creazione come

$$t_{\text{creazione}} = t_c + (t_s - t_i).$$

Il pregio di questa formulazione è che un eventuale scarto costante dell'orologio del dispositivo compare in entrambe le marche del client e si elide nel calcolo: ciò che rimane è l'istante di creazione riportato nel tempo del server, a meno del solo ritardo di trasmissione, tipicamente trascurabile. Un piccolo margine di tolleranza impedisce inoltre che eventuali imprecisioni collochino la creazione nel futuro del server. In questo modo l'ordine cronologico delle attività resta affidabile anche quando l'invio avviene in differita.

Per contenere la dimensione delle registrazioni e mantenere tempi di trascrizione gestibili, la durata di ogni nota vocale è limitata a pochi minuti. Tale scelta, oltre a riflettere la natura delle annotazioni operative (descrizioni brevi e puntuali, non conversazioni estese), mantiene i file ampiamente entro i limiti dimensionali accettati dal servizio di trascrizione.

4.5 Configurazione della trascrizione

La trascrizione è eseguita interamente lato server. Una volta caricato l'audio su Dropbox, il backend ne invia una copia all'endpoint di trascrizione dell'API di OpenAI (`/v1/audio/transcriptions`) tramite una richiesta HTTP `multipart`, con un timeout di 120 secondi. Il modello utilizzato è `whisper-1`, corrispondente alla variante *large-v2* di Whisper. La chiamata trasmette unicamente due parametri: il modello e la lingua, impostata esplicitamente su italiano (`language=it`). Non vengono passati né un `prompt`, né una `temperature` personalizzata, né uno specifico formato di risposta: l'output viene quindi restituito nel formato JSON predefinito, dal quale il backend estrae il solo campo testuale. La trascrizione è considerata riuscita soltanto se la risposta ha esito positivo e il campo testuale non è vuoto; in caso contrario il tentativo è considerato fallito e, come visto nella Sezione 4.3, l'interazione viene marcata in *errore*.

La trascrizione viene tentata automaticamente solo quando l'interazione contiene un audio e non è già accompagnata da un testo fornito dall'utente; in caso contrario il testo esistente viene mantenuto senza interrogare il modello. L'unica forma di condizionamento è il vincolo sulla lingua: non viene fornita alcuna guida lessicale o contestuale. È significativo notare che l'API metterebbe a disposizione, proprio attraverso il parametro `prompt` discusso nel Capitolo 2 (limitato agli ultimi 224 token per `whisper-1`), l'unico gancio di adattamento leggero compatibile con un uso ospitato; la scelta progettuale è, tuttavia, di non utilizzarlo. Si tratta di un utilizzo *zero-shot* puro, coerente con la filosofia del modello descritta nel Capitolo 2 e con la decisione, discussa nel Capitolo 3, di non ricorrere ad alcuna tecnica di adattamento di dominio.

4.6 Trascrizione, correzione e affidabilità

La trascrizione prodotta da Whisper è trattata come un livello modificabile e non come un dato definitivo: come descritto nella sezione sul modello dei dati, l'output grezzo del modello viene conservato come trascrizione originale, le correzioni generano nuove versioni senza sovrascrivere l'originale ed è sempre possibile ripristinarlo. Questa impostazione fa sì che un eventuale errore del modello non comprometta l'informazione, che resta verificabile riascoltando l'audio.

A valle della trascrizione è applicata una sola forma di elaborazione automatica: un filtro che individua e scarta gli output corrispondenti a formule tipiche di sottotitoli o titoli di coda (ad esempio crediti del tipo "sottotitoli a cura di…" o stringhe ricorrenti come quelle della comunità Amara.org), notoriamente generate da Whisper in presenza di audio silenzioso o fortemente rumoroso. Quando una trascrizione corrisponde a tali formule, essa viene annullata e l'interazione marcata come errore, lasciando comunque disponibile l'audio originale. Questo filtro non fornisce alcuna guida lessicale al riconoscimento, ma mitiga uno specifico caso di allucinazione, fenomeno analizzato più in dettaglio nel Capitolo 5.

4.7 Collaborazione e verifica nella filiera

Il brogliaccio non è un archivio individuale, ma un canale condiviso tra gli attori che ruotano attorno a una stessa azienda agricola. Ogni interazione è associata a un'azienda ed è visibile agli utenti che vi hanno accesso, secondo il loro ruolo. I ruoli previsti comprendono l'agricoltore (titolare dell'azienda), il terzista, il delegato, il consulente e il tecnico, oltre a ruoli di servizio e amministrativi.

Ogni interazione ha un autore e può avere un destinatario esplicito; consulenti, tecnici e amministratori possono indirizzare un'interazione a uno specifico destinatario oppure diffonderla a tutti gli attori dell'azienda, e possono inoltre registrare un'interazione per conto di un agricoltore o di un terzista. Questo schema realizza concretamente la funzione di collaborazione e verifica tra gli attori di filiera richiamata nell'introduzione: una nota vocale trascritta diventa un oggetto condiviso che più soggetti possono consultare, integrare e correggere prima che la relativa informazione confluisca, indirettamente, nel QDCA.

La verifica è regolata da meccanismi di autorizzazione coerenti con i ruoli. L'autore di un'interazione e gli amministratori possono sempre gestirla; un consulente o un tecnico possono correggere — ma non eliminare — le interazioni prodotte da un agricoltore o da un terzista. Anche in questo caso la correzione preserva l'originale ed è reversibile, così che la verifica non comporti la perdita del dato di partenza.

Il sistema tiene, inoltre, traccia dello stato di lettura delle interazioni e notifica gli attori interessati, evitando che le informazioni rilevanti vengano trascurate.

L'accesso alle informazioni è infine circoscritto per azienda: la visibilità di ciascun utente è determinata dalle aziende a cui è associato e dal ruolo che vi ricopre, e l'autenticazione dei client avviene tramite token. In questo modo il brogliaccio resta accessibile soltanto agli attori legittimamente coinvolti nella gestione di una determinata azienda.

4.8 Condizioni operative e rilevanza per l'analisi

Il sistema opera in condizioni non controllate. Le registrazioni avvengono in campo aperto, possono contenere rumore ambientale, parlato spontaneo, esitazioni e interruzioni, e ricorrono frequentemente a terminologia tecnica: nomi commerciali di prodotti, riferimenti a trattamenti, colture e operazioni agronomiche. Sono proprio queste condizioni a rendere il dominio agricolo un banco di prova significativo per un modello ASR generalista utilizzato in modalità *zero-shot*.

Il caso di studio così delineato fornisce gli elementi necessari all'analisi critica del capitolo successivo, in cui le scelte progettuali descritte — l'uso di Whisper tramite API senza adattamento di dominio, la conservazione dell'audio come fonte primaria, l'assenza di guida lessicale — vengono confrontate con quanto emerso dallo stato dell'arte.

Capitolo 5

Analisi critica

Questo capitolo mette in relazione diretta il caso di studio *Traccial'Abilità* con quanto emerso nello stato dell'arte, con l'obiettivo di valutare in modo critico l'efficacia dell'utilizzo di Whisper tramite API in un contesto reale. L'analisi considera la trascrizione non come fonte primaria dell'informazione, ma come strumento di semplificazione dell'elaborazione: la registrazione audio rimane sempre associata al testo e costituisce il riferimento principale.

5.1 Coerenza con il modello di Radford

L'utilizzo di Whisper nel sistema *Traccial'Abilità* rispecchia pienamente la filosofia del modello proposta da Radford et al.[1], ovvero l'impiego in modalità *zero-shot* senza fine-tuning specifico.

Il sistema sfrutta infatti il modello tramite API, senza accesso diretto ai pesi o possibilità di adattamento supervisionato. Questo approccio è coerente con l'idea di utilizzare modelli *foundation* robusti e generalizzabili in contesti reali. Nel caso specifico, però, il modello non assume il ruolo di certificatore dell'informazione: produce una rappresentazione testuale utile all'elaborazione, mentre la certezza dell'origine del dato rimane garantita dalla conservazione dell'audio registrato dall'operatore.

5.2 Punti di forza osservati nel caso di studio

Dall'analisi del sistema emergono diversi aspetti positivi, che conviene distinguere in base a ciò che è direttamente osservabile dall'architettura e a ciò che, in assenza di una valutazione quantitativa sul campo, è invece atteso sulla base della letteratura.

Sono direttamente verificabili dal sistema due aspetti:

- **facilità di integrazione**: l'uso dell'API consente una rapida implementazione senza necessità di un'infrastruttura di machine learning dedicata;
- **tracciabilità della fonte**: l'associazione persistente tra audio e trascrizione, garantita anche dall'hash crittografico, consente di mantenere il riferimento all'origine dell'informazione anche quando il testo generato necessita di correzione.

Sono invece attese sulla base dei risultati di Radford et al.[1] e della letteratura successiva, ma non misurate direttamente in questo lavoro:

- una **buona qualità generale della trascrizione**, che in condizioni acustiche favorevoli dovrebbe fornire una base utile alla consultazione e alla successiva elaborazione;
- una ragionevole **robustezza al parlato spontaneo** e non strutturato, tipico del contesto agricolo.

Poiché nel caso di studio non è stata condotta una valutazione sistematica del WER su audio reale, queste due ultime proprietà vanno intese come aspettative fondate sulla robustezza generale documentata per il modello, non come misure proprie del sistema.

A titolo illustrativo, e senza alcuna pretesa di rappresentatività statistica, la Tabella 5.1 riporta alcune trascrizioni prodotte su audio reale del sistema. L'Esempio 1 mostra il caso favorevole, pur restando un esempio e non una misura: in condizioni acustiche pulite e senza accento marcato la trascrizione è sostanzialmente perfetta, e perfino il nome commerciale *SULFAN* è riconosciuto correttamente; le sole differenze rispetto al riferimento sono di formattazione (minuscole anziché maiuscole).

A questi due piani di evidenza — ciò che è direttamente osservabile dall'architettura e ciò che è atteso dalla letteratura — se ne aggiunge un terzo, intermedio: il sistema è in esercizio, e i dati d'uso aggregati relativi al periodo di osservazione (dicembre 2025 – giugno 2026) descrivono come è stato effettivamente utilizzato. Si tratta di una descrizione dell'uso reale, non di una misura controllata della qualità della trascrizione; entro questo limite, alcuni di essi corroborano le assunzioni progettuali discusse nel Capitolo 4. La Tabella 5.2 ne riassume i valori principali.

Da questi dati si possono trarre alcune osservazioni, distinguendo con cura ciò che essi mostrano in modo solido da ciò che consentono soltanto di ipotizzare. In primo luogo, confermano che il sistema è realmente in uso e a una scala non trascurabile: oltre milleduecento interazioni prodotte da trentacinque utenti su altrettante aziende, con una netta prevalenza della modalità vocale (65,7%), a riprova della cen-

Tabella 5.1: Esempi illustrativi di trascrizione su audio reale (esempi, non una valutazione quantitativa). In grassetto gli errori di riconoscimento; le differenze di sola formattazione (maiuscole/minuscole e forma dei numeri) non sono evidenziate.

Riferimento (pronunciato)	**Trascrizione di Whisper**
In data 21 febbraio 2026 abbiamo concimato il grano e l'orzo con il SULFAN 250 chili ettaro circa 3.750 chili totali.	in data 21 febbraio 2026 abbiamo concimato il grano e l'orzo con il sulfan 250 chili ettaro circa 3.750 chili totali.
14 giugno 2026 distribuito su ettari 6,2 di barbabietole da zucchero, miscela di Univoq litri 1,5 per ettaro, poltiglia dispersiva, chilogrammi 2,5 per ettaro, Kostar, chilogrammi 0,75 per ettaro, nitrato potassico, chilogrammi 4 per ettaro, con tre quintali di acqua per ettaro.	14 giugno 2026 distribuito su ettari 6,2 di barbabietole da zucchero, miscela di **Univoc** litri 1,5 per ettaro, poltiglia dispersiva, chilogrammi 2,5 per ettaro, Kostar, chilogrammi 0,75 per ettaro, nitrato potassico, chilogrammi 4 per ettaro, con tre quintali di acqua per ettaro.
Primo marzo 2026 distribuito concime Labin 8-5-15, presemina Barbabietole 6,2 ettari, 3,22 quintali per ettaro distribuiti.	Primo marzo **duemilaventiseistribuito con Cime** Labin, otto, cinque, quindici, **presemino** Barbabietole, sei virgola due ettari, tre virgola ventidue quintali per ettaro distribuiti.

tralità della trascrizione automatica nell'uso effettivo del sistema. In secondo luogo, la durata media di una nota vocale — circa 17 secondi — conferma empiricamente la natura delle annotazioni descritta nella Sezione 4.4: descrizioni brevi e puntuali, non conversazioni estese. Ne consegue che, per questo caso d'uso, il problema della trascrizione *long-form* discusso nel Capitolo 2 risulta in larga parte marginale, poiché le registrazioni rientrano ampiamente nella finestra gestita nativamente dal modello.

In terzo luogo, la quota di note vocali in stato *errore* è contenuta al 3,2%. Poiché tale stato aggrega, come descritto nella Sezione 4.5, sia i fallimenti della chiamata all'API sia gli annullamenti operati dal filtro anti-formule (Sezione 4.6), questo valore costituisce un *limite superiore* alla frequenza di attivazione del filtro discusso nella Sezione 5.3.4: la mitigazione delle allucinazioni da audio silenzioso interviene dunque, al più, su una frazione ridotta delle trascrizioni.

Una cautela particolare merita infine il tasso di correzione. Sull'intero periodo il 32,6% delle note vocali risulta corretto manualmente, valore che nei mesi più recenti e a maggior volume si stabilizza attorno al 20%. È forte la tentazione di leggere questo dato come un indicatore di qualità — una quota contenuta di correzioni suggerirebbe che l'output sia "abbastanza buono" da non richiedere intervento nella maggioranza

Tabella 5.2: Dati d'uso aggregati del sistema nel periodo di osservazione (dicembre 2025 – giugno 2026). Le percentuali di stato sono calcolate sulle sole note vocali. I valori descrivono l'uso reale del sistema e non costituiscono una valutazione quantitativa della qualità della trascrizione.

Indicatore	Valore
Interazioni totali	1.248
utenti / aziende	35 / 35
Note vocali	820 (65,7%)
Note testuali	428 (34,3%)
Durata media nota vocale	∼17 s
Stato delle note vocali	
trascritto	527 (64,3%)
corretto	267 (32,6%)
errore	26 (3,2%)
in attesa	0 (0%)

dei casi — ma una simile lettura va trattata con prudenza, per almeno tre ragioni. Anzitutto, una correzione non equivale a un errore di trascrizione: l'operatore può modificare il testo per aggiungere informazione, riformulare il proprio parlato o adeguarne la forma, sicché il tasso sovrastima la frequenza degli errori del modello. In secondo luogo, non tutti gli errori vengono necessariamente corretti, per cui il dato non costituisce nemmeno un limite superiore pulito. Infine, l'andamento nel tempo è confuso da fattori d'uso — la crescita del volume e il progressivo spostamento verso le note testuali — e non è quindi interpretabile come una qualità che migliora o peggiora. Il tasso di correzione va perciò inteso come un indicatore operativo debole e condizionato, coerente con — ma non dimostrativo di — l'ipotesi che la trascrizione fornisca nella maggioranza dei casi una base utilizzabile senza intervento; non è in alcun modo un sostituto del WER, la cui assenza è discussa nella Sezione 5.6.

5.3 Limiti emersi nel contesto agricolo

Accanto ai punti di forza richiamati, il contesto agricolo evidenzia alcune criticità significative: alcune intrinseche al modello, altre proprie dello scenario basato su API.

5.3.1 Riconoscimento di termini tecnici e nomi di prodotto

Il limite più rilevante, e quello su cui la letteratura converge con maggior forza, riguarda il riconoscimento dei termini specialistici. È un comportamento coerente

con i lavori su *jargon injection*[4] e *rare-word recognition*[20], e soprattutto con il benchmark agricolo[11], in cui Whisper — nella stessa versione esposta dall'API — mostra una marcata fragilità proprio sul lessico di dominio.

Nel contesto agricolo, la presenza di nomi commerciali, terminologia agronomica e abbreviazioni può portare a errori sistematici: il decoder, che si comporta come un modello linguistico condizionato dall'audio, tende a sostituire il termine raro con una parola più frequente foneticamente simile. Tali errori non compromettono la conservazione dell'informazione originaria, poiché l'audio rimane sempre disponibile, ma possono ridurre l'efficacia della trascrizione come strumento di supporto all'elaborazione del QDCA.

L'Esempio 2 della Tabella 5.1 mostra questo limite nella forma più nitida: in una registrazione per il resto trascritta correttamente — quantità in cifre, coltura e altri prodotti — l'unico errore cade sul nome commerciale *Univoq*, reso "Univoc". Il confronto con l'Esempio 1 è istruttivo: il concime *SULFAN*, presumibilmente più diffuso, viene riconosciuto, mentre il fungicida *Univoq*, più recente e di nicchia, no. La fragilità, cioè, non colpisce indistintamente i nomi commerciali, ma quelli meno frequenti, coerentemente con il meccanismo di sostituzione verso le forme più probabili descritto sopra.

5.3.2 Accento, dialetto e variazione linguistica

Una caratteristica del caso di studio non ancora messa a fuoco riguarda la varietà linguistica del parlato in ingresso. Gli operatori agricoli che utilizzano il sistema presentano, in misura variabile, un accento regionale marcato e, in alcuni casi, possono ricorrere a lessico dialettale, specie per la terminologia agronomica locale. È utile interpretare questa condizione alla luce dei lavori discussi nel Capitolo 3, distinguendo però tre situazioni che il linguaggio comune tende a confondere: l'accento, il dialetto e la lingua a basse risorse.

Un accento regionale su italiano standard costituisce in primo luogo una sfida di natura acustica: il lessico e la grammatica restano quelli dello standard, mentre varia la realizzazione fonetica. Questo è precisamente l'asse su cui Whisper è documentato come robusto[1], e l'italiano è una lingua a risorse relativamente elevate, ben rappresentata nei dati di addestramento; l'API ospitata espone inoltre una variante di grandi dimensioni (large-v2), superiore ai modelli piccoli valutati sull'italiano in[17]. Per il solo accento, dunque, un'analogia con varietà linguistiche minoritarie come il tedesco svizzero sovrastimerebbe la difficoltà: ci si attende che il modello gestisca bene il contenuto generale anche in presenza di accenti marcati. L'Esempio 3 della Tabella 5.1 illustra concretamente questo punto: l'accento incide sulla segmentazione — l'anno e il verbo fusi in "duemilaventiseistribuito", "concime" diviso in "con

Cime" — e sulla morfologia ("presemino" per "presemina"), mentre il nome commerciale *Labin*, la coltura e tutti i valori numerici restano corretti; gli errori cadono cioè sulla forma e non sull'informazione agronomica critica.

Questa attesa positiva sul versante linguistico è supportata dai dati. Nel corpus di addestramento di Whisper l'italiano dispone di circa 2.585 ore di trascrizioni[1]: una lingua ben rappresentata — assai più di quelle a basse risorse esaminate in [11] — pur collocandosi nella fascia medio-alta della distribuzione e lontano dal vertice occupato da inglese, cinese, tedesco e spagnolo. Radford et al. [1] documentano una forte correlazione ($r^2 = 0{,}83$) tra la quantità di dati per lingua e il WER *zero-shot*, con un dimezzamento dell'errore per ogni aumento di sedici volte dei dati; gli scarti peggiori riguardano lingue con sistemi di scrittura unici e tipologicamente distanti, condizione che non si applica all'italiano. Coerentemente, in condizioni acustiche pulite Whisper large-v2 raggiunge sull'italiano un WER intorno all'8% [13], a fronte di circa il 37% del più piccolo Whisper Base sullo stesso dato. Il limite per il caso di studio non risiede quindi nella lingua in sé, ma nel lessico di dominio e nelle condizioni di registrazione.

Il ricorso a un dialetto vero e proprio si avvicina invece maggiormente alla situazione studiata in [16] per il tedesco svizzero, una varietà in rapporto di diglossia con lo standard. Il meccanismo lì osservato è che Whisper non trascrive il dialetto in modo letterale, ma lo normalizza, producendo di fatto una traduzione verso la varietà standard ad alte risorse. Due differenze rendono però l'analogia solo parziale. La prima è linguistica: tedesco svizzero e tedesco standard sono sistematicamente molto vicini e il modello beneficia dell'ampia disponibilità di dati di tedesco standard, mentre i dialetti italiani variano in misura maggiore nella distanza dallo standard e sono assai meno rappresentati. La seconda, più rilevante per il caso di studio, riguarda la direzione della normalizzazione: nello scenario svizzero la resa in tedesco standard è l'output desiderato, mentre nel contesto agricolo la normalizzazione di un termine locale o dialettale verso una forma standard più comune può cancellare proprio l'informazione utile — il nome di una varietà colturale, di una pratica o di un prodotto. Lo stesso meccanismo che in [16] è un pregio diventa qui un rischio.

Il filo conduttore che attraversa i tre regimi è la tendenza, documentata in [11;10], a sostituire i termini di dominio rari con parole più frequenti foneticamente simili. Il fenomeno non dipende dall'accento o dal dialetto in quanto tali, ma dalla rarità del termine, ed è semmai amplificato dal dialetto e dalla scarsa copertura linguistica. È un meccanismo che opera anche sull'italiano ad alte risorse: in [20] Whisper sull'italiano mostra un WER generale contenuto a fronte di un errore sensibilmente più alto

[1] Dato del pannello *Multilingual Speech Recognition* della Figura 11 (Appendice E, *Training Dataset Statistics*) di Radford et al. [1]; la stessa figura riporta per l'italiano 2.145 ore di dati di traduzione, qui non rilevanti.

sulle parole fuori vocabolario, a conferma dello stesso schema — buone prestazioni sul parlato comune, fragilità sui termini rari.

Da queste considerazioni discendono due avvertenze. La prima è metodologica: nessuno dei lavori citati misura italiano agricolo accentato o dialettale —[16] riguarda il tedesco svizzero,[17] l'italiano letto in condizioni controllate,[11] lingue indiane,[20] l'italiano in dominio generale — sicché le valutazioni qui proposte costituiscono una triangolazione ragionata da evidenze adiacenti e non una misura diretta sul caso di studio. La seconda è progettuale e riconduce alla scelta centrale del sistema: proprio perché l'output del modello su lessico locale e dialettale è incerto, la conservazione della registrazione audio come fonte primaria non è un dettaglio implementativo, ma la salvaguardia che rende gestibile tale incertezza, consentendo in ogni momento di verificare e correggere il testo a fronte di un termine normalizzato o sostituito.

5.3.3 Assenza di contextual biasing

Il sistema non applica alcuna forma di guida lessicale al riconoscimento. È però importante inquadrare correttamente questo limite. Il *contextual biasing* in senso proprio — come negli approcci di CB-Whisper[5] e dei lavori affini[8] — agisce sulle probabilità dei token, sulle rappresentazioni dell'encoder o sugli stati interni del modello, ed è quindi strutturalmente non disponibile attraverso l'API ospitata, che non espone alcuno di questi elementi. Più che una scelta omessa, l'assenza di *contextual biasing* è dunque una conseguenza dello scenario di accesso adottato. L'unico gancio di guida lessicale effettivamente disponibile — l'inserimento di terminologia nel prompt testuale (Capitolo 2) — non viene attualmente utilizzato, ed è la principale direzione di miglioramento leggero discussa nella sezione successiva.

5.3.4 Allucinazioni e affidabilità

In casi rari, il modello produce testo plausibile ma non presente nell'audio, fenomeno noto come allucinazione, analizzato in *Careless Whisper*[12]. Lo studio rileva che circa l'1% delle trascrizioni contiene intere frasi allucinate e che tali allucinazioni si concentrano in modo sproporzionato in corrispondenza di lunghe porzioni audio non vocali, cioè in presenza di silenzio o rumore prolungato.

Questo aspetto è particolarmente critico in un contesto operativo. Nel caso di *Traccial'Abilità*, l'associazione tra audio e trascrizione riduce il rischio di perdita dell'informazione, ma non elimina la necessità di controllare il testo quando esso viene utilizzato per elaborazioni successive o per la compilazione assistita del QDCA.

Un caso particolare di allucinazione osservabile con Whisper riguarda la generazione di formule tipiche di sottotitoli o titoli di coda (ad esempio crediti del tipo "sotto-

titoli a cura di. . . ” o stringhe ricorrenti come quelle della comunità Amara.org) in presenza di audio silenzioso o fortemente rumoroso. Per questo specifico fenomeno il sistema *Traccial'Abilità* adotta una contromisura esplicita: un filtro a lista di esclusione che riconosce tali formule e annulla la trascrizione, marcando l'interazione come errore. Questa contromisura intercetta proprio la condizione individuata in [12] — l'audio con lunghe porzioni non vocali — e la sua efficacia fa sì che, nel sistema in esercizio, tali formule non permangano nel testo conservato: la condizione che le genera viene riconosciuta e l'interazione marcata come errore prima che l'output spurio raggiunga il brogliaccio. Si tratta di una mitigazione mirata e a basso costo, applicata in fase di post-elaborazione e quindi compatibile con l'uso tramite API; essa non risolve però il problema più generale delle allucinazioni su contenuti plausibili, per le quali resta determinante la conservazione dell'audio come riferimento verificabile.

5.3.5 Vincoli operativi delle API

L'utilizzo tramite API introduce ulteriori limitazioni operative:

- l'impossibilità di effettuare *fine-tuning*, completo o del solo decoder, come nei lavori [18;19;22];
- l'assenza di controllo sul processo di decodifica e sui parametri interni;
- la dipendenza da latenza, disponibilità del servizio e costi per chiamata.

Questi vincoli non sono difetti del sistema, ma caratteristiche dello scenario di accesso scelto, e definiscono il perimetro entro cui sono praticabili le tecniche di adattamento discusse nella sezione seguente.

5.4 Tecniche della letteratura applicabili al caso

Alcune delle tecniche analizzate nello stato dell'arte risultano potenzialmente applicabili anche nel contesto *Traccial'Abilità*, ma la loro adozione va valutata tenendo conto di un'avvertenza preliminare sulla trasferibilità dei risultati riportati in letteratura.

Gli studi citati non valutano tutti la stessa variante di Whisper: alcuni impiegano modelli di piccole dimensioni [17], altri versioni recenti come large-v3 [16], mentre lo studio più pertinente al dominio agricolo [11] adotta proprio large-v2 attraverso l'API, ossia il modello utilizzato in questo lavoro. Poiché un modello più grande parte in genere da un WER di base inferiore, l'entità assoluta del miglioramento ottenibile con una data tecnica non è necessariamente la stessa misurata in quegli studi. Sarebbe però errato concluderne che le tecniche siano semplicemente meno

efficaci su large-v2: per il riconoscimento di parole rare — il problema centrale del caso di studio — il limite dipende soprattutto dalla copertura del termine nei dati di addestramento e dalla distribuzione di frequenza appresa, più che dalla dimensione del modello, sicché il margine di miglioramento resta ampio anche partendo da large-v2, come mostra la fragilità di tale modello sul lessico agricolo documentata in [11]. Lo studio [20] osserva anzi miglioramenti relativi maggiori proprio sui dataset in cui Whisper partiva da prestazioni migliori. La trasferibilità quantitativa precisa di ciascuna tecnica a large-v2 resta dunque una questione aperta, che la sola letteratura disponibile non consente di risolvere; le considerazioni che seguono vanno lette in questa chiave qualitativa.

5.4.1 Tecniche realisticamente adottabili

Tra le tecniche realisticamente adottabili rientrano:

- **prompting e contesto testuale**: l'inserimento di termini di dominio nel breve prompt testuale messo a disposizione dall'API (Capitolo 2), unico gancio di condizionamento compatibile con un uso ospitato;
- **descrizioni generate da LLM**: la generazione automatica di una descrizione del contesto da fornire nel prompt, secondo l'idea proposta in [10];
- **dizionari di termini**: la costruzione e l'aggiornamento di un glossario di dominio da iniettare come testo, soluzione ispirata alla *keyword-guided adaptation* [7], il cui metodo completo richiede però addestramento.

Queste tecniche non richiedono modifica del modello e sono compatibili con l'uso tramite API. Va però ricordato, coerentemente con il Capitolo 3, che il prompt di whisper-1 è limitato (al massimo 224 token) e va inteso come suggerimento e non come vincolo: si tratta dunque di meccanismi di adattamento dichiaratamente deboli.

Per rendere concreto questo meccanismo conviene mostrare che cosa conterrebbe, in pratica, un simile prompt e quanto lessico vi possa entrare. Un prompt di adattamento per il caso di studio potrebbe assumere la forma di una breve descrizione contestuale seguita da un glossario dei termini critici, ad esempio:

> *Nota operativa agricola per il quaderno di campagna. Possibili termini di dominio: SULFAN, Univoq, Kostar, Labin, poltiglia dispersiva, nitrato potassico, barbabietola da zucchero, presemina, principio attivo, quintali per ettaro.*

Il vincolo dei 224 token di `whisper-1` (Capitolo 2) è meno stringente di quanto possa sembrare. I termini rari — proprio quelli che il modello tende a sbagliare

— si frammentano in più sottounità: con il tokenizzatore di `large-v2`, ad esempio, *SULFAN* occupa quattro token (`S/UL/F/AN`) e *Univoq* tre. Anche tenendo conto di questa frammentazione, un glossario di una quindicina di voci di dominio occupa circa settanta token, e il budget disponibile consente quindi dell'ordine di quarantacinque-cinquanta voci: sufficienti, nella pratica, a coprire i nomi commerciali, le varietà e i termini agronomici critici di una singola azienda. Il prompt di esempio riportato sopra, che unisce descrizione e glossario, ne impiega circa settantacinque, lasciando ampio margine. Resta inteso, coerentemente con il Capitolo 3, che tale prompt agisce come suggerimento e non come vincolo: l'inserimento dei termini ne aumenta la probabilità di corretto riconoscimento, ma non la garantisce.

5.4.2 Tecniche non direttamente applicabili

Altre tecniche risultano invece non direttamente applicabili al contesto considerato:

- il *fine-tuning* supervisionato, completo o del solo decoder [18;19;22];
- l'adattamento con dati sintetici [21];
- il *contextual biasing* che interviene sulla decodifica o sugli stati interni del modello [5;6;8];
- l'apprendimento di prompt continui (*prompt tuning*) [23] e il condizionamento su esempi di parlato [9], che presuppongono rispettivamente l'addestramento del modello e una modalità di input (coppie audio-trascrizione) non supportata dall'endpoint di trascrizione;
- le architetture modificate, come Whisper-TAD [15].

Queste tecniche richiedono accesso ai pesi, al processo di decodifica o agli stati interni del modello, oppure infrastrutture di addestramento non disponibili in uno scenario basato esclusivamente sull'API di inferenza ospitata.

5.5 Sintesi critica

Dall'analisi emergono alcune conclusioni. Whisper rappresenta una soluzione efficace e pragmatica per affiancare una trascrizione automatica all'audio registrato in contesti reali, coerente con la filosofia *zero-shot* del modello. Le principali limitazioni riguardano il trattamento del lessico di dominio, amplificato — ma non causato — dall'eventuale variazione di accento o dialetto degli operatori; si tratta di un limite atteso dalla letteratura e in larga parte intrinseco all'uso generalista del modello, più che una carenza del sistema. La conservazione dell'audio come fonte primaria, garantita dall'hash crittografico e dal versionamento non distruttivo, permette di

gestire gli errori e le allucinazioni della trascrizione senza mai perdere il riferimento all'informazione originaria. La letteratura propone numerose soluzioni di adattamento, ma esse si dispongono lungo uno spettro di accesso al modello e solo quelle agenti sul prompt testuale sono compatibili con uno scenario basato esclusivamente sull'API; per giunta, la loro efficacia quantitativa su large-v2 non è deducibile con certezza dai risultati pubblicati, ottenuti su varianti del modello differenti.

In questo senso il caso *Traccial'Abilità* si colloca in una posizione intermedia tra l'utilizzo pratico e immediato della tecnologia e la possibilità di un miglioramento incrementale attraverso tecniche leggere di adattamento, da validare però sul campo prima di poterne quantificare il beneficio.

5.6 Validità e limiti dell'analisi

L'analisi condotta in questo capitolo è, per scelta, di natura qualitativa. Per rendere espliciti i confini entro cui le sue conclusioni si mantengono — non per indebolirle, ma per delimitarle — è utile raccogliere in un unico punto le cautele metodologiche già richiamate lungo il capitolo.

- **Assenza di una valutazione quantitativa per scelta.** Il lavoro non comprende un benchmark del WER su audio reale: gli esempi della Tabella 5.1 hanno valore illustrativo e non statistico, e i dati d'uso della Sezione 5.2 descrivono l'utilizzo reale del sistema senza misurarne la qualità di trascrizione. Questa scelta riflette l'assenza, nel contesto considerato, di un corpus agricolo annotato con trascrizioni di riferimento, e l'orientamento dichiaratamente qualitativo della tesi.
- **Indicatore operativo, non metrica di qualità.** Il tasso di correzione introdotto nella Sezione 5.2 è un indicatore operativo debole e condizionato — da editing non riconducibili a errori, da errori non sempre corretti e da variazioni nella composizione d'uso — e non va inteso come un proxy quantitativo del WER.
- **Caso di studio singolo.** Le osservazioni si riferiscono a una specifica realtà operativa: l'insieme delle aziende, le colture, i prodotti e le caratteristiche linguistiche degli operatori riflettono un contesto particolare. La generalizzabilità ad altri scenari agricoli, con altre varietà colturali o altre aree dialettali, va quindi assunta con cautela.
- **Triangolazione da evidenze adiacenti.** Come osservato nella Sezione 5.3.2, nessuno dei lavori citati misura direttamente l'italiano agricolo accentato o dialettale: le valutazioni linguistiche proposte combinano evidenze vicine —

il tedesco svizzero[16], l'italiano letto in condizioni controllate[17], le lingue indiane in contesto agricolo[11], l'italiano in dominio generale[20] — ma non costituiscono una misura diretta sul caso di studio.

- **Tecniche proposte ma non validate.** Le tecniche di adattamento leggero discusse nella Sezione 5.4.1 sono argomentate come compatibili con l'uso tramite API, ma non sono state implementate né misurate all'interno del sistema: la loro efficacia nel contesto specifico resta una previsione fondata sulla letteratura, non un risultato.
- **Trasferibilità a `large-v2` come questione aperta.** Come anticipato in apertura della Sezione 5.4, i lavori citati impiegano varianti eterogenee di Whisper; l'entità quantitativa dei benefici ottenibili sulla variante esposta dall'API (`whisper-1`, ossia `large-v2`) non è deducibile con certezza dai risultati pubblicati.

Nessuna di queste delimitazioni intacca l'impianto qualitativo dell'analisi: esse circoscrivono ciò che il presente lavoro può affermare e, al tempo stesso, rendono esplicite le precondizioni di una sua estensione su base quantitativa, oggetto della sezione seguente.

5.7 Direzioni di sviluppo

La direzione di sviluppo più naturale consiste nell'agire sull'unica leva di adattamento compatibile con l'uso tramite API — il prompt testuale (Capitolo 2) — e nell'osservarne l'effetto attraverso la telemetria già disponibile nel sistema.

Sul piano dell'intervento, si tratterebbe di costruire e mantenere un glossario di dominio — i nomi commerciali dei prodotti effettivamente impiegati dalle aziende, le varietà colturali e i termini agronomici locali — e di iniettarlo come testo nel prompt, eventualmente nella forma di una breve descrizione contestuale generata automaticamente sull'idea proposta in[10]. Come mostrato concretamente nella Sezione 5.4.1, il budget di 224 token consente dell'ordine di alcune decine di voci di glossario, sufficienti a coprire il lessico critico di un'azienda: l'intervento è quindi non solo compatibile con l'API, ma anche realizzabile entro i suoi vincoli.

Sul piano della valutazione, e coerentemente con la natura qualitativa del lavoro, l'effetto dell'introduzione del prompt potrebbe essere osservato come una variazione del tasso di correzione operativo (Sezione 5.2) prima e dopo l'intervento, su volumi d'uso comparabili. Va però ribadito che si tratterebbe di un primo segnale operativo, soggetto alle stesse cautele illustrate nella Sezione 5.6 — in particolare ai fattori che ne confondono l'andamento — e non di una misura controllata: una sua attribu-

zione netta richiederebbe di controllare la composizione d'uso nei due periodi. Una quantificazione rigorosa del beneficio — idealmente tramite una metrica pesata sui termini di dominio, nello spirito dell'AWWER proposto in[11], calcolata su un set di riferimento — costituirebbe il riferimento ideale, ma esula dagli obiettivi di questo lavoro.

Un'estensione collaterale, infine, riguarda la telemetria stessa: distinguere, all'interno dello stato *errore*, gli annullamenti operati dal filtro anti-formule dai fallimenti della chiamata all'API consentirebbe di trasformare il limite superiore riportato nella Sezione 5.2 in una misura diretta della frequenza di attivazione del filtro, dando sostanza quantitativa all'analisi delle allucinazioni della Sezione 5.3.4.

Conclusioni

La tesi ha analizzato l'utilizzo del modello Whisper in un contesto reale, evidenziando punti di forza e limiti della trascrizione automatica applicata alle attività agricole.

Nel caso di *Traccial'Abilità*, Whisper non viene utilizzato per sostituire la fonte originaria dell'informazione, ma per affiancare all'audio registrato una rappresentazione testuale utile alla consultazione e all'elaborazione del Quaderno di Campagna dell'Agricoltore. La conservazione dell'audio insieme alla trascrizione permette di mantenere la certezza dell'origine dell'informazione e di verificare eventuali errori del modello.

L'analisi indica che Whisper rappresenta una soluzione efficace e facilmente integrabile, ma che ulteriori miglioramenti sono possibili attraverso tecniche di adattamento di dominio, in particolare per il riconoscimento di terminologia tecnica, nomi commerciali, colture e prodotti agricoli. Tecniche leggere come *prompting*, dizionari di dominio e descrizioni contestuali generate automaticamente risultano particolarmente promettenti perché compatibili con l'uso tramite API, pur richiedendo una validazione sul campo per quantificarne l'effettivo beneficio.

In conclusione, l'adozione di Whisper in *Traccial'Abilità* rappresenta una scelta pragmatica e coerente con lo stato dell'arte: consente di ridurre il carico manuale dell'operatore e di rendere più accessibili le informazioni raccolte in campo, mantenendo però l'audio originale come riferimento primario e garantendo la possibilità di controllo e correzione del testo trascritto.

Bibliografia

[1] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever. *Robust Speech Recognition via Large-Scale Weak Supervision*. Proceedings of the 40th International Conference on Machine Learning (ICML), PMLR 202, pp. 28492–28518, 2023. arXiv:2212.04356.

[2] OpenAI. *Speech to Text*. Documentazione API di OpenAI. `https://platform.openai.com/docs/guides/speech-to-text` (consultato il 24 giugno 2026).

[3] OpenAI. *Introducing ChatGPT and Whisper APIs*. 1 marzo 2023. `https://openai.com/index/introducing-chatgpt-and-whisper-apis/` (consultato il 24 giugno 2026).

[4] M.-T. Nguyen, P.-D. Nguyen, T.-H. Luu, X.-Q. Nguyen, T.-D. Nguyen, and J. Yang. *Improving Speech Recognition with Jargon Injection*. Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL), pp. 490–499, 2024.

[5] Y. Li, Y. Li, M. Zhang, C. Su, J. Yu, M. Piao, X. Qiao, M. Ma, Y. Zhao, and H. Yang. *CB-Whisper: Contextual Biasing Whisper using Open-Vocabulary Keyword-Spotting*. Proceedings of LREC-COLING 2024, pp. 2941–2946, 2024.

[6] A. Mittal, S. Sarawagi, P. Jyothi, G. Saon, and G. Kurata. *Speech-enriched Memory for Inference-time Adaptation of ASR Models to Word Dictionaries*. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 14820–14835, 2023.

[7] A. Shamsian, A. Navon, N. Glazer, G. Hetz, and J. Keshet. *Keyword-Guided Adaptation of Automatic Speech Recognition*. Proceedings of Interspeech 2024, pp. 732–736, 2024. doi:10.21437/Interspeech.2024-391.

[8] V. Lall and Y. Liu. *Contextual Biasing to Improve Domain-specific Custom Vocabulary Audio Transcription without Explicit Fine-Tuning of Whisper Model*. arXiv:2410.18363, 2024.

[9] S. Wang, C.-H. Yang, J. Wu, and C. Zhang. *Can Whisper Perform Speech-Based In-Context Learning?* Proceedings of the 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 13421–13425, 2024. doi:10.1109/ICASSP48485.2024.10446502.

[10] J. Suh, I. Na, and W. Jung. *Improving Domain-Specific ASR with LLM-Generated Contextual Descriptions*. arXiv:2407.17874, 2024.

[11] Chandrashekar M. S., V. Singh, and L. Pedapudi. *Benchmarking Automatic Speech Recognition for Indian Languages in Agricultural Contexts*. arXiv:2602.03868, 2026.

[12] A. Koenecke, A. S. G. Choi, K. X. Mei, H. Schellmann, and M. Sloane. *Careless Whisper: Speech-to-Text Hallucination Harms*. Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency (FAccT '24), 2024. doi:10.1145/3630106.3658996.

[13] S. Katkov, A. Liotta, and A. Vietti. *Benchmarking Whisper Under Diverse Audio Transformations and Real-Time Constraints*. In A. Karpov and V. Delić (eds.), *Speech and Computer* (SPECOM 2024), Lecture Notes in Artificial Intelligence, vol. 15299, pp. 82–91. Springer, 2025.

[14] D. Macháček, R. Dabre, and O. Bojar. *Turning Whisper into Real-Time Transcription System*. Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the ACL (IJCNLP-AACL): System Demonstrations, pp. 17–24, 2023.

[15] C. Lavigne and A. Stasica. *Whisper-TAD: A General Model for Transcription, Alignment and Diarization of Speech*. Proceedings of CLIB 2024 (Computational Linguistics in Bulgaria), 2024.

[16] E. L. Dolev, C. F. Lutz, and N. Aepli. *Does Whisper Understand Swiss German? An Automatic, Qualitative and Human Evaluation*. Proceedings of the Eleventh Workshop on NLP for Similar Languages, Varieties, and Dialects (VarDial), pp. 28–40, 2024.

[17] M. Quadrini, M. Loreti, M. Luciani, I. Corradetti, M. Carboni, and S. Vagnoni. *A Comparative Study of Speech-to-Text for Italian*. Proceedings of Ital-IA 2025: 5th National Conference on Artificial Intelligence, Trieste, 2025.

[18] V. Timmel, C. Paonessa, M. Vogel, D. Perruchoud, and R. Kakooe. *Fine-tuning Whisper on Low-Resource Languages for Real-World Applications*. Proceedings of the Swiss Text Analytics Conference (SwissText), 2025.

[19] B. Kurian, A. Upadhyay, and A. Sengupta. *Domain-Specific Adaptation for ASR through Text-Only Fine-Tuning*. Proceedings of MMLoSo 2025: First Workshop on Multimodal Models for Low-Resource Contexts and Social Impact, pp. 78–85, 2025.

[20] Y. Jogi, V. Aggarwal, S. S. Nair, Y. Verma, and A. Kubba. *Improving Rare-Word Recognition of Whisper in Zero-Shot Settings*. Proceedings of the 2024 IEEE Spoken Language Technology Workshop (SLT), Macao, pp. 216–223, 2024. doi:10.1109/SLT61566.2024.10832199.

[21] M. Tran, Y. Pang, D. Paul, L. Pandey, K. Jiang, J. Guo, K. Li, S. Zhang, X. Zhang, and X. Lei. *A Domain Adaptation Framework for Speech Recognition Systems with Only Synthetic Data*. Proceedings of ICASSP 2025 – 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Hyderabad, India, pp. 1–5, 2025. doi:10.1109/ICASSP49660.2025.10887883.

[22] Y. Liu, X. Yang, and D. Qu. *Exploration of Whisper Fine-tuning Strategies for Low-Resource ASR*. EURASIP Journal on Audio, Speech, and Music Processing, 2024:29, 2024. doi:10.1186/s13636-024-00349-3.

[23] Y. Chen, W. Zhang, H. Zhang, X. Yang, and D. Qu. *Exploring the Potential of Prompting Methods in Low-Resource Speech Recognition with Whisper*. NLPCC 2024, Lecture Notes in Artificial Intelligence, vol. 15361, pp. 382–393. Springer, 2025.