UNIVERSITÀ POLITECNICA DELLE MARCHE

FACOLTÀ DI INGEGNERIA

Corso di Laurea Magistrale in
Ingegneria Informatica e dell'Automazione

Progettazione e valutazione di una pipeline RAG per l'interrogazione di manuali tecnici nel dominio yacht di lusso

Design and evaluation of a RAG pipeline for querying technical manuals in the luxury yacht domain

Relatore:
Prof. Mancini Adriano

Laureando:
De Ritis Riccardo

Anno Accademico 2025-2026

Sommario

Il presente lavoro di tesi affronta il problema dell'accesso alla documentazione tecnica nel settore degli yacht di lusso, caratterizzata da manuali in formato PDF complessi, eterogenei e spesso articolati in procedure distribuite su più pagine. In tale contesto, la consultazione manuale dei documenti può risultare lenta e difficoltosa, soprattutto quando le informazioni necessarie riguardano condizioni preliminari, avvertenze di sicurezza, passaggi operativi e riferimenti a sezioni differenti.

L'obiettivo del lavoro è stato progettare, sviluppare e valutare una pipeline basata sul paradigma della Retrieval-Augmented Generation per l'interrogazione di manuali tecnici mediante domande formulate in linguaggio naturale. La soluzione proposta integra una fase di ingestion documentale, un sistema di indicizzazione e retrieval e una fase di generazione della risposta, con particolare attenzione alla conservazione del contesto e alla tracciabilità delle fonti.

Sono state confrontate diverse strategie di preprocessing e chunking, tra cui la pipeline aziendale iniziale, un approccio basato su Unstructured, una strategia fondata sugli artefatti prodotti da MinerU2.5 e una pipeline personalizzata sviluppata durante il tirocinio presso SailADV. La soluzione proposta utilizza una rappresentazione gerarchica parent/child, combina ricerca semantica e lessicale, applica tecniche di fusione e reranking dei risultati e fornisce al modello generativo un contesto più ampio e verificabile rispetto ai frammenti recuperati.

La valutazione sperimentale è stata condotta su un dataset costruito manualmente a partire da otto manuali tecnici, composto da 70 domande associate alle rispettive procedure, ai contenuti di riferimento e alle pagine di provenienza. I risultati mostrano un miglioramento significativo rispetto alla baseline aziendale: la pipeline personalizzata raggiunge una Page Recall pari a 0.9443 e una Retrieval F1 pari a 0.9072, ottenendo inoltre i valori più elevati di Retrieval Precision, Retrieval Recall e BERTScore F1 tra le configurazioni considerate. La fase generativa mostra punteggi medi elevati in termini di correttezza, fedeltà al contesto e completezza, confermando la capacità del sistema di produrre risposte tecniche aderenti alla documentazione disponibile.

Nel complesso, il lavoro dimostra la fattibilità di una pipeline RAG modulare e osservabile per supportare la consultazione di manuali tecnici di bordo. La soluzione non sostituisce la documentazione ufficiale né le competenze del personale tecnico, ma rappresenta uno strumento di supporto volto a ridurre i tempi di ricerca, migliorare l'accesso alle informazioni rilevanti e mantenere il collegamento con le fonti originali.

Abstract

This thesis addresses the problem of accessing technical documentation in the luxury yacht domain, where PDF manuals are often complex, heterogeneous and organized into procedures spanning multiple pages. In this context, manually searching for relevant information can be time-consuming and difficult, especially when the required answer depends on preliminary conditions, safety warnings, operational steps and references distributed across different sections.

The aim of the work was to design, develop and evaluate a Retrieval Augmented Generation pipeline for querying technical manuals through natural language questions. The proposed system includes a document ingestion stage, an indexing and retrieval component, and a response generation stage, with particular attention to context preservation and source traceability.

Several preprocessing and chunking strategies were compared, including the initial company pipeline, an approach based on Unstructured, a strategy relying on artifacts produced by MinerU2.5, and a custom pipeline developed during the internship at SailADV. The proposed solution adopts a hierarchical parent/child representation, combines semantic and lexical retrieval, applies result fusion and reranking techniques, and provides the generative model with a broader and more verifiable context than the individual retrieved fragments.

The experimental evaluation was conducted on a manually built dataset derived from eight technical manuals and composed of 70 questions associated with the corresponding procedures, reference contents and source pages. The results show a significant improvement over the initial company baseline: the custom pipeline achieves a Page Recall of 0.9443 and a Retrieval F1 of 0.9072, while also obtaining the highest Retrieval Precision, Retrieval Recall and BERTScore F1 among the evaluated configurations. The generation stage also achieves high average scores in correctness, faithfulness and completeness, confirming the system's ability to produce technical answers grounded in the available documentation.

Overall, the work demonstrates the feasibility of a modular and observable RAG pipeline to support the consultation of onboard technical manuals. The proposed solution is not intended to replace official documentation or the expertise of technical personnel, but rather to serve as a support tool aimed at reducing search time, improving access to relevant information and preserving the connection with the original sources.

Indice

Capitolo 1

Introduzione

Il presente lavoro di tesi, svolto nell'ambito di un tirocinio presso SailADV, ha riguardato la progettazione, lo sviluppo e la valutazione di una pipeline basata sul paradigma della Retrieval-Augmented Generation (RAG), finalizzata all'interrogazione di manuali tecnici relativi al settore degli yacht di lusso. Il progetto si colloca nel più ampio processo di digitalizzazione e valorizzazione della documentazione tecnica e mira a rendere più semplice, rapido e affidabile l'accesso alle informazioni contenute in documenti PDF caratterizzati da strutture complesse ed eterogenee.

Il presente capitolo introduce il contesto applicativo nel quale è stato sviluppato il progetto, presenta l'azienda ospitante e descrive le principali motivazioni alla base del lavoro. Vengono inoltre delineati gli obiettivi generali del sistema, successivamente approfonditi attraverso l'analisi dei fondamenti teorici, delle scelte architetturali e delle strategie adottate per l'elaborazione e l'indicizzazione dei documenti. I capitoli conclusivi sono invece dedicati alla metodologia di valutazione sperimentale, all'analisi dei risultati ottenuti e alla discussione dei principali limiti e dei possibili sviluppi futuri della soluzione proposta.

1.1 Contesto del lavoro

Negli ultimi anni, la crescente digitalizzazione dei processi industriali ha determinato un aumento significativo della quantità di dati e documenti prodotti durante le diverse fasi del ciclo di vita di sistemi e prodotti complessi. Accanto ai dati acquisiti mediante sensori, prove sperimentali e sistemi di monitoraggio, una parte rilevante della conoscenza tecnica continua a essere conservata all'interno di documenti testuali, quali manuali d'uso, guide di installazione, procedure di manutenzione, specifiche tecniche e rapporti di collaudo.

Tali documenti costituiscono una fonte essenziale di conoscenza operativa. Al loro interno sono descritte le modalità di utilizzo degli apparati, le procedure da seguire in condizioni ordinarie o di emergenza, gli intervalli di manutenzione, le configurazioni ammesse e le precauzioni necessarie a garantire il funzionamento sicuro dei sistemi. La possibilità di accedere rapidamente a queste informazioni assume pertanto un ruolo rilevante sia durante le attività di progettazione e installazione, sia nelle successive fasi di esercizio, manutenzione e diagnostica.

Il problema della gestione della documentazione tecnica risulta particolarmente evidente nei settori caratterizzati dalla presenza di sistemi eterogenei e fortemente interconnessi. In tali contesti, le informazioni necessarie allo svolgimento di una determinata operazione possono essere distribuite tra documenti prodotti da fornitori differenti e redatti secondo strutture, terminologie e convenzioni non uniformi. La ricerca delle informazioni non consiste quindi soltanto nell'individuazione di una parola o di una frase, ma richiede spesso la ricostruzione del contesto nel quale una determinata indicazione risulta applicabile.

Il settore dello yachting, in particolare quello degli yacht di grandi dimensioni e di alta gamma, rappresenta un esempio significativo di tale complessità. Uno yacht moderno può essere considerato un sistema tecnologico articolato, all'interno del quale coesistono apparati meccanici, elettrici, elettronici, idraulici e informatici. Il corretto funzionamento dell'imbarcazione dipende dall'interazione tra numerosi sottosistemi, tra cui motori e sistemi di propulsione, generatori, pompe, stabilizzatori, impianti di climatizzazione, sistemi per il trattamento e la distribuzione dell'acqua, dispositivi di automazione e apparati per il monitoraggio delle condizioni operative.

Ciascun componente può essere fornito da un produttore differente ed essere corredato da una propria documentazione tecnica. Di conseguenza, il patrimonio documentale disponibile a bordo o presso le strutture incaricate della gestione dell'imbarcazione può comprendere un numero elevato di manuali, spesso redatti in lingua inglese e distribuiti in vari formati. La consultazione di tali documenti coinvolge diverse figure professionali, tra cui progettisti, tecnici, responsabili di commessa, personale di bordo, manutentori e operatori impegnati nelle attività di collaudo e verifica.

La complessità non dipende esclusivamente dal numero dei documenti disponibili. I manuali tecnici presentano frequentemente una struttura articolata, composta da capitoli, sezioni, sottosezioni, elenchi numerati, tabelle, figure e avvertenze di sicurezza. Le informazioni relative a una singola operazione possono essere distribuite su più pagine oppure contenere riferimenti ad altre parti del documento. Una procedura può, ad esempio, essere preceduta da condizioni preliminari, includere avvertenze che ne condizionano l'esecuzione ed essere seguita da verifiche da effettuare al termine dell'intervento. Una corretta interpretazione richiede quindi di preservare le relazioni esistenti tra questi diversi elementi.

Gli strumenti tradizionali di ricerca testuale consentono di individuare le occorrenze di determinate parole, ma non sempre permettono di identificare il contenuto maggiormente rilevante rispetto a una richiesta formulata in linguaggio naturale. Uno stesso termine può comparire in procedure differenti, mentre un'operazione può essere descritta nel manuale mediante espressioni diverse da quelle utilizzate dall'utente. Inoltre, il recupero di una singola frase può risultare insufficiente quando, per comprenderne correttamente il significato, è necessario disporre dell'intera procedura, delle condizioni preliminari e delle relative avvertenze.

I recenti sviluppi nel campo dei Large Language Models (LLM) hanno reso possibile la realizzazione di sistemi in grado di interpretare domande formula-

te in linguaggio naturale e di generare risposte articolate. Tuttavia, l'impiego diretto di un modello linguistico non garantisce che le informazioni prodotte siano effettivamente presenti nella documentazione di riferimento, né che risultino corrette rispetto allo specifico apparato considerato. In un dominio tecnico, una risposta plausibile ma non aderente al contenuto dei manuali può risultare poco affidabile e potenzialmente problematica, soprattutto quando riguarda procedure operative o indicazioni di sicurezza.

Per integrare le capacità dei modelli linguistici con una base documentale specifica è possibile adottare un'architettura di tipo Retrieval-Augmented Generation. In un sistema RAG, la richiesta dell'utente viene utilizzata per ricercare e selezionare le porzioni dei documenti ritenute maggiormente pertinenti. I contenuti recuperati vengono successivamente forniti al modello linguistico come contesto per la generazione della risposta. Tale approccio consente di favorire la produzione di risposte fondate sulla documentazione disponibile e di mantenere un collegamento con le fonti originali, riducendo il ricorso alla sola conoscenza interna del modello.

L'efficacia di un sistema RAG dipende tuttavia in larga misura dalle modalità con cui i documenti vengono analizzati, estratti, suddivisi e indicizzati. Porzioni di testo eccessivamente estese possono contenere informazioni non pertinenti e ridurre la precisione del recupero; frammenti troppo piccoli possono invece separare una procedura dalle condizioni preliminari o dalle avvertenze necessarie alla sua corretta interpretazione. La rappresentazione della struttura e del contenuto dei documenti costituisce pertanto una componente centrale dell'intera pipeline.

Il presente lavoro si colloca in questo contesto e riguarda l'applicazione delle tecniche RAG alla consultazione di manuali relativi ad apparati impiegati nel settore degli yacht di lusso. L'attività, svolta nell'ambito di un tirocinio presso SailADV, ha preso in esame documentazione riferita a diverse tipologie di componenti, tra cui motori, pompe, boiler, ventilatori, sistemi idrici pressurizzati e stabilizzatori. Il dominio applicativo considerato consente di analizzare concretamente le problematiche legate all'estrazione dei contenuti dai documenti PDF, alla conservazione della struttura delle procedure e al recupero delle informazioni necessarie per rispondere alle richieste degli utenti.

1.2 L'azienda SailADV

Figura 1.1: Logo dell'azienda ospitante: SailADV. Fonte: [1]

SailADV Group è un'azienda indipendente di ingegneria e tecnologia attiva nel settore dello yachting e, più in generale, in quello marittimo. La società integra competenze ingegneristiche, attività di misurazione e analisi dei dati, servizi tecnici e ricerca applicata, offrendo supporto nelle diverse fasi del ciclo di vita di uno yacht. La natura multidisciplinare del gruppo consente di affrontare le problematiche relative alle imbarcazioni considerando congiuntamente gli aspetti progettuali, prestazionali, operativi e tecnologici.

L'attività dell'azienda si fonda sulla valorizzazione dei dati prodotti durante le operazioni tecniche, con l'obiettivo di affiancare all'esperienza degli operatori strumenti di analisi quantitativa. Tale impostazione è sintetizzata da SailADV nel principio secondo cui ciò che può essere misurato può essere analizzato e, di conseguenza, migliorato.

La misurazione rappresenta pertanto uno degli elementi centrali del metodo adottato dall'azienda. I dati raccolti durante prove, controlli e attività operative non vengono considerati esclusivamente come risultati isolati, ma come informazioni da organizzare, contestualizzare e interpretare. Questo processo consente di descrivere in modo oggettivo il comportamento di un'imbarcazione e dei suoi apparati, contribuendo alla verifica delle prestazioni e all'analisi di fenomeni complessi. Secondo l'impostazione dichiarata dall'azienda, tale approccio mira a trasformare le attività operative in informazioni strutturate e i dati in strumenti di supporto ai processi decisionali.

Un'ulteriore caratteristica dell'azienda è rappresentata dall'integrazione tra competenze ingegneristiche tradizionali e tecnologie digitali. Le conoscenze maturate nell'ambito delle prove e delle misurazioni sono affiancate da strumenti per l'acquisizione, la gestione e l'analisi dei dati, nonché da applicazioni basate sull'intelligenza artificiale. Le informazioni provenienti dai diversi apparati presenti a bordo possono così essere raccolte e analizzate nell'ambito di una visione complessiva dell'imbarcazione. Lo yacht viene infatti considerato come un sistema complesso, costituito da numerosi sottosistemi interconnessi, il cui comportamento deve essere valutato tenendo conto delle effettive condizioni operative [1].

1.3 Motivazioni e obiettivi

La motivazione principale del presente lavoro deriva dalla frammentazione e dalla complessità della documentazione tecnica impiegata nel settore nautico. Le imbarcazioni moderne integrano numerosi apparati, spesso realizzati da produttori differenti e corredati da manuali specifici. Il patrimonio documentale disponibile può pertanto essere costituito da un insieme esteso ed eterogeneo di file, organizzati secondo strutture, terminologie e convenzioni grafiche non uniformi.

Le informazioni necessarie allo svolgimento di una determinata operazione non sono sempre concentrate in un'unica sezione del manuale. Una procedura può estendersi su più pagine e includere condizioni preliminari, avvertenze di sicurezza, riferimenti ad altri capitoli e verifiche da eseguire al termine dell'intervento. Di conseguenza, l'individuazione del contenuto corretto richiede

spesso una consultazione manuale approfondita e una conoscenza preliminare dell'organizzazione del documento.

Tale esigenza assume particolare rilevanza per le figure professionali che devono consultare la documentazione durante le attività operative, tra cui comandanti, membri dell'equipaggio, tecnici di cantiere e responsabili della gestione della flotta. In questi contesti, la rapidità con cui viene individuata una procedura può incidere sui tempi necessari per diagnosticare un problema, eseguire un intervento o ripristinare il funzionamento di un apparato. Un accesso più diretto alle informazioni tecniche può pertanto contribuire a ridurre i tempi di ricerca e, nei casi di guasto o manutenzione, a limitare i periodi di indisponibilità dell'imbarcazione o del sistema interessato.

Gli strumenti tradizionali di ricerca testuale consentono di individuare parole o espressioni presenti nei documenti, ma richiedono generalmente che l'utente conosca la terminologia adottata dal costruttore. Inoltre, i risultati di una ricerca per parole chiave possono essere costituiti da singole occorrenze prive del contesto necessario alla corretta interpretazione di una procedura. L'esigenza non consiste quindi soltanto nell'individuare un passaggio semanticamente affine alla domanda, ma nel recuperare la sezione appropriata insieme alle condizioni, alle avvertenze e ai passaggi operativi che ne completano il significato.

A partire da tali considerazioni, l'obiettivo generale del lavoro è stato progettare, sviluppare e valutare una pipeline di Retrieval-Augmented Generation (RAG) per l'interrogazione di manuali tecnici in formato PDF relativi al settore degli yacht di lusso. Il sistema è stato concepito per ricevere una domanda in linguaggio naturale, individuare all'interno dei documenti i contenuti pertinenti e generare una risposta fondata sulle informazioni recuperate, mantenendo un collegamento con le pagine e le sezioni di provenienza.

Un primo obiettivo specifico ha riguardato l'analisi delle modalità con cui i manuali vengono trasformati in contenuti indicizzabili. In particolare, sono state confrontate differenti strategie di parsing, preprocessing e chunking, al fine di studiare l'influenza esercitata dalla rappresentazione documentale sulla qualità del retrieval. Sono stati considerati sia approcci basati su strumenti e librerie esistenti, sia una pipeline personalizzata, sviluppata con l'obiettivo di riconoscere la struttura dei manuali e preservare unità procedurali significative.

Un secondo obiettivo è consistito nell'individuazione di una strategia capace di bilanciare la precisione del recupero e la completezza del contesto fornito al modello generativo. Il sistema deve infatti recuperare porzioni sufficientemente specifiche rispetto alla domanda, evitando al contempo di separare i singoli passaggi dalle condizioni preliminari e dalle avvertenze necessarie alla loro corretta interpretazione. A tale scopo è stata approfondita una rappresentazione gerarchica di tipo parent/child, nella quale la ricerca viene effettuata su frammenti più granulari, mentre alla fase di generazione vengono fornite sezioni più ampie e contestualizzate.

Un ulteriore obiettivo ha riguardato la progettazione delle fasi di indicizzazione e retrieval. Il lavoro ha esaminato la combinazione tra ricerca semantica e ricerca lessicale, l'impiego di tecniche di fusione dei risultati e il successivo riordinamento dei contenuti recuperati mediante reranking. Tale analisi è stata finalizzata a migliorare l'individuazione delle procedure sia quando la doman-

da contiene i termini tecnici presenti nel manuale, sia quando viene formulata utilizzando espressioni differenti.

Il progetto ha inoltre previsto la definizione di una metodologia sperimentale per il confronto tra le soluzioni analizzate. A tale scopo è stato costruito manualmente un dataset di valutazione composto da domande associate alle rispettive procedure, ai contenuti di riferimento e alle pagine dei manuali nelle quali tali informazioni sono presenti. Il dataset ha consentito di valutare separatamente la capacità del sistema di recuperare le pagine corrette, la corrispondenza tra il contenuto recuperato e quello atteso e la qualità della risposta generata.

Gli obiettivi principali del lavoro possono pertanto essere sintetizzati nei seguenti punti:

1. progettare e sviluppare una pipeline completa per l'estrazione, l'indicizzazione e l'interrogazione di manuali tecnici in formato PDF;
2. confrontare differenti strategie di parsing, preprocessing e chunking;
3. sviluppare una rappresentazione documentale in grado di preservare il contesto delle procedure;
4. realizzare un sistema di retrieval basato sulla combinazione di segnali semantici e lessicali;
5. generare risposte tecniche aderenti ai contenuti dei manuali e corredate da riferimenti alle fonti;
6. costruire un dataset controllato e definire metriche per la valutazione delle fasi di retrieval e generazione;
7. individuare i principali vantaggi e limiti delle strategie analizzate.

Il lavoro non è stato quindi orientato principalmente alla realizzazione di un'interfaccia conversazionale, ma allo studio e alla valutazione dell'intera catena di elaborazione che conduce dal documento originale alla risposta finale. L'attenzione è stata posta, in particolare, sul rapporto tra la qualità dell'estrazione e della rappresentazione documentale, l'efficacia del recupero delle informazioni e l'affidabilità del contenuto generato.

Capitolo 2

Stato dell'arte

Il presente capitolo introduce i principali concetti teorici e tecnologici alla base del sistema sviluppato. In particolare, vengono descritti i Large Language Models (LLM) e il paradigma della Retrieval-Augmented Generation (RAG), evidenziandone il ruolo nella realizzazione di sistemi per l'accesso e l'interrogazione di basi documentali specialistiche.

Vengono inoltre illustrate le principali tecniche di elaborazione documentale impiegate per estrarre, strutturare e suddividere i contenuti dei file PDF, nonché i metodi di Information Retrieval utilizzati per individuare le informazioni pertinenti rispetto a una richiesta formulata dall'utente. La trattazione comprende infine i database vettoriali, impiegati per memorizzare, indicizzare e ricercare le rappresentazioni vettoriali dei contenuti.

Tali elementi costituiscono il quadro teorico di riferimento per comprendere le scelte architetturali e implementative adottate nella progettazione e nello sviluppo della pipeline descritta nei capitoli successivi.

2.1 Large Language Models

I Large Language Models (LLM) sono modelli di apprendimento automatico progettati per elaborare e generare contenuti espressi in linguaggio naturale. Addestrati su grandi quantità di dati testuali, apprendono regolarità statistiche e relazioni contestuali tra gli elementi del linguaggio. Tali capacità consentono loro di svolgere una pluralità di compiti, tra cui la generazione e la sintesi di testi, la traduzione automatica e la risposta a domande formulate dagli utenti.

La maggior parte degli LLM moderni è basata sull'architettura Transformer, introdotta nel 2017. Uno dei suoi elementi fondamentali è il meccanismo di attenzione, attraverso il quale il modello determina il contributo dei diversi token alla rappresentazione di ciascun elemento della sequenza. Rispetto alle precedenti architetture ricorrenti, i Transformer consentono di elaborare in parallelo i token durante l'addestramento e di modellare efficacemente le dipendenze tra elementi anche distanti all'interno del testo [2].

Il processo di addestramento comprende generalmente una fase iniziale di pre-training, durante la quale il modello apprende a predire elementi del testo sulla base del contesto disponibile. Nel caso dei modelli linguistici generativi au-

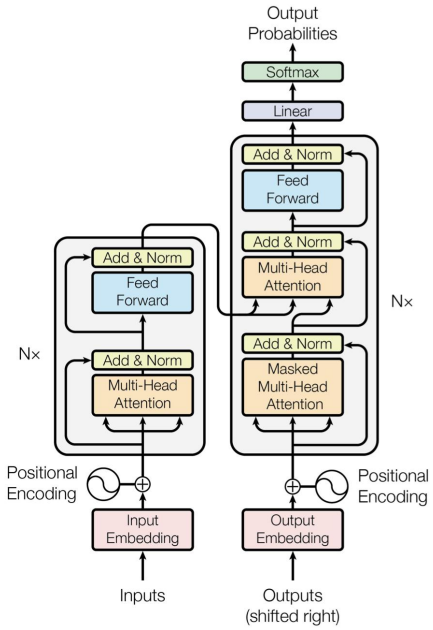


Figura 2.1: Architettura generale del Transformer. Il modello elabora una sequenza di token attraverso blocchi di encoder e decoder basati sul meccanismo di self-attention, che consente di modellare le relazioni tra elementi della sequenza anche quando si trovano in posizioni distanti. Questa architettura costituisce la base di molti Large Language Models moderni. Fonte: [2].

toregressivi, l'obiettivo consiste tipicamente nella previsione del token successivo della sequenza [3]. Attraverso l'esposizione a un ampio insieme di dati, il modello apprende regolarità linguistiche e incorpora informazioni presenti nei dati di addestramento. Successivamente, il modello può essere sottoposto a procedure di fine-tuning e instruction tuning, finalizzate ad adattarne il comportamento a compiti specifici e a migliorarne la capacità di seguire le istruzioni formulate dagli utenti.

Durante l'inferenza, il modello riceve in ingresso un prompt che può comprendere la richiesta dell'utente, le istruzioni operative ed eventuali informazioni di contesto. La risposta viene generata in modo autoregressivo: a ogni passaggio, il modello calcola una distribuzione di probabilità sui possibili token successivi, dalla quale viene selezionato un token secondo la strategia di decodifica adottata. Il processo viene ripetuto utilizzando anche i token già generati fino al raggiungimento di una condizione di arresto. Tale meccanismo consente di produrre testi linguisticamente fluidi e coerenti, ma non garantisce la correttezza fattuale o la verificabilità delle informazioni generate [4].

Un LLM può infatti produrre affermazioni plausibili dal punto di vista linguistico, ma inesatte, non supportate dalle fonti o incoerenti con il contesto. Tale fenomeno viene comunemente indicato con il termine *allucinazione*. Il modello può inoltre non disporre di informazioni specialistiche, riservate o successive al periodo a cui risalgono i dati utilizzati per il suo addestramento. Questi limiti risultano particolarmente rilevanti nella consultazione della documentazione tecnica, poiché una risposta deve essere coerente con le istruzioni fornite dallo

specifico costruttore e non può fondarsi esclusivamente sulle conoscenze generali incorporate nel modello.

Per applicare gli LLM a domini specialistici è quindi opportuno fornire informazioni contestuali provenienti da fonti pertinenti e affidabili. Una delle principali modalità adottate a tale scopo è la Retrieval-Augmented Generation, nella quale la generazione della risposta viene supportata da contenuti recuperati dinamicamente da una base documentale. Questo paradigma, approfondito nella sezione successiva, consente di integrare le capacità linguistiche degli LLM con le informazioni contenute nei documenti di interesse.

2.2 Retrieval-Augmented Generation

La Retrieval-Augmented Generation (RAG) è un paradigma che integra le capacità generative dei Large Language Models con un sistema di recupero dell'informazione da una base documentale esterna. In questo approccio, il modello linguistico può utilizzare contenuti recuperati dinamicamente da una collezione di documenti durante il processo di generazione della risposta, invece di basarsi esclusivamente sulle informazioni apprese durante l'addestramento.

In una tipica architettura RAG, la richiesta formulata dall'utente viene inizialmente elaborata da un componente di retrieval, il cui compito è individuare i contenuti maggiormente pertinenti all'interno della collezione documentale. Gli elementi recuperati vengono successivamente inseriti nel contesto del modello linguistico insieme alla domanda originale e alle eventuali istruzioni operative. Il modello genera quindi la risposta utilizzando le informazioni fornite dal sistema di ricerca, oltre alle capacità linguistiche e alle conoscenze incorporate nei propri parametri [5].

Il funzionamento di una pipeline RAG può essere suddiviso in due fasi principali. Durante la fase offline, i documenti vengono acquisiti, elaborati, suddivisi in unità indicizzabili e memorizzati all'interno di uno o più indici. Nella fase online, invece, la richiesta dell'utente viene utilizzata per interrogare tali indici, recuperare i contenuti rilevanti e costruire il contesto da fornire al modello generativo. Tra il recupero e la generazione possono inoltre essere inserite operazioni di fusione, filtraggio e riordinamento dei risultati.

Il recupero delle informazioni può essere effettuato mediante tecniche lessicali, dense o ibride. I metodi lessicali, come BM25, stimano la rilevanza dei documenti sulla base della corrispondenza tra i termini della query e quelli presenti nei contenuti indicizzati, tenendo conto anche della loro frequenza e distribuzione [7]. I metodi di dense retrieval rappresentano invece query e documenti mediante vettori numerici, denominati embedding, e ne valutano la similarità nello spazio vettoriale [8]. Le due famiglie di approcci possono essere combinate per considerare sia le corrispondenze terminologiche esatte sia le relazioni semantiche tra espressioni differenti.

Rispetto all'impiego isolato di un LLM, un RAG consente di utilizzare informazioni specialistiche, private o aggiornabili senza intervenire direttamente sui parametri del modello. La base documentale può infatti essere modificata indipendentemente dal componente generativo e i contenuti recuperati possono

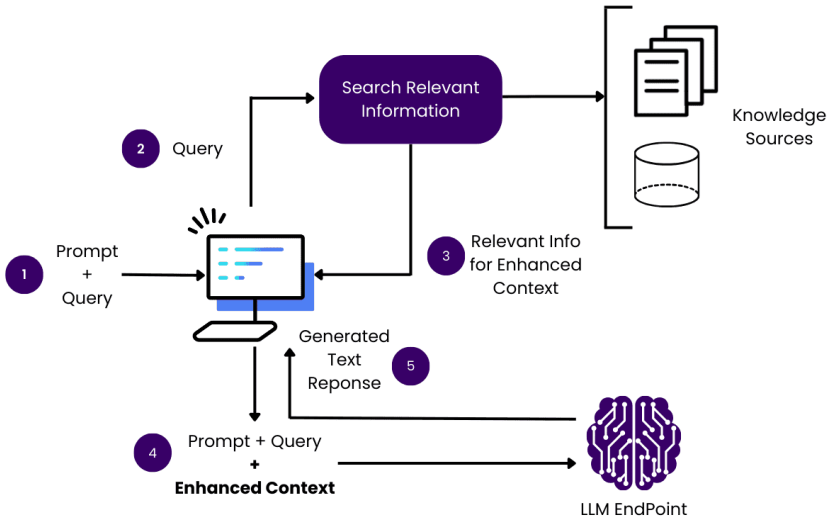


Figura 2.2: Schema generale di una pipeline Retrieval-Augmented Generation. La query dell'utente viene utilizzata per recuperare informazioni rilevanti da una base documentale esterna; tali informazioni vengono poi fornite al modello linguistico come contesto aggiuntivo per la generazione della risposta. Fonte: [6]

essere associati alle rispettive fonti. Ciò rende questo paradigma particolarmente adatto ai contesti nei quali le risposte devono essere fondate su documenti specifici e verificabili [5].

L'impiego di una pipeline RAG può inoltre ridurre la probabilità che il modello generi informazioni non supportate, poiché la risposta viene orientata dal contesto recuperato. Tale beneficio dipende tuttavia dalla qualità dell'intera catena di elaborazione. Se i contenuti selezionati sono errati, incompleti, ridondanti o poco pertinenti, anche la risposta finale può risultare inaccurata. Analogamente, la presenza delle informazioni corrette nella base documentale non garantisce che esse vengano recuperate, né che il modello le utilizzi correttamente durante la generazione.

Nel caso della documentazione tecnica, il retrieval non deve limitarsi a individuare porzioni semanticamente affini alla domanda, ma deve recuperare un contesto sufficiente a preservare le condizioni preliminari, le avvertenze e la sequenza dei passaggi che compongono una procedura. La qualità complessiva di un sistema RAG dipende pertanto dall'interazione tra elaborazione documentale, strategia di suddivisione del testo, rappresentazione dei contenuti, indicizzazione, recupero del contesto e generazione della risposta.

2.3 Tecniche di elaborazione documentale

L'elaborazione documentale comprende l'insieme delle tecniche impiegate per trasformare un documento digitale in una rappresentazione strutturata e utilizzabile da un sistema informatico. Nel contesto delle pipeline RAG, tale fase precede l'indicizzazione e ha lo scopo di estrarre, organizzare e suddividere i con-

tenuti dei documenti, preservandone, per quanto possibile, la struttura logica e il significato.

L'estrazione dei contenuti dai file PDF presenta diverse difficoltà. Il formato PDF è stato progettato principalmente per preservare la rappresentazione visiva delle pagine nei diversi ambienti di visualizzazione e stampa. Di conseguenza, la disposizione grafica percepita dal lettore non corrisponde necessariamente a una struttura logica esplicitamente rappresentata all'interno del file [9]. Il testo può essere memorizzato come un insieme di caratteri o blocchi posizionati mediante coordinate, senza informazioni sufficienti per identificare direttamente titoli, paragrafi, colonne e sezioni.

Gli strumenti tradizionali di parsing estraggono il testo digitale insieme a proprietà quali la posizione, la dimensione e lo stile dei caratteri. Tali informazioni possono essere utilizzate per ricostruire righe e blocchi testuali e per riconoscere elementi strutturali come titoli, paragrafi ed elenchi. Questo approccio risulta generalmente efficace per documenti semplici, caratterizzati da un layout regolare e da un ordine di lettura prevalentemente lineare.

Nei PDF più complessi, tuttavia, la sola estrazione testuale può produrre risultati incoerenti. Layout a più colonne, tabelle, immagini, schemi e riquadri possono rendere ambiguo l'ordine con cui i contenuti devono essere interpretati. Un parser che non consideri adeguatamente la disposizione spaziale degli elementi può, ad esempio, alternare righe provenienti da colonne differenti oppure separare i valori di una tabella dalle rispettive intestazioni. La ricostruzione dell'ordine di lettura costituisce pertanto un aspetto fondamentale dell'analisi documentale [10].

Un caso differente è rappresentato dai documenti acquisiti mediante scansione, nei quali il testo non è disponibile come contenuto digitale direttamente estraibile, ma è incorporato nell'immagine della pagina. In tali situazioni è necessario ricorrere a tecniche di Optical Character Recognition (OCR), finalizzate a individuare le regioni testuali e a convertire i caratteri presenti nell'immagine in testo elaborabile [11]. L'accuratezza del riconoscimento può essere influenzata da diversi fattori, tra cui la risoluzione e la qualità della scansione, l'orientamento della pagina, le caratteristiche tipografiche e la presenza di elementi grafici.

L'impiego dell'OCR non risolve automaticamente il problema della ricostruzione della struttura documentale. Anche quando le singole parole vengono riconosciute correttamente, è necessario stabilire a quale paragrafo, colonna, tabella, figura o didascalia appartengano. Inoltre, gli errori introdotti durante il riconoscimento possono propagarsi alle fasi successive, alterando termini tecnici, valori numerici, unità di misura o codici identificativi particolarmente rilevanti ai fini della ricerca.

Per affrontare tali difficoltà vengono utilizzate tecniche di analisi del layout, che considerano congiuntamente il contenuto testuale e la posizione degli elementi all'interno della pagina. L'obiettivo è individuare regioni omogenee, classificare le diverse tipologie di contenuto e ricostruire le relazioni tra titoli, paragrafi, elenchi, tabelle, immagini e didascalie. Queste informazioni consentono di ottenere una rappresentazione maggiormente aderente alla struttura logica del documento rispetto a una semplice sequenza di caratteri estratti.

Negli ultimi anni si sono inoltre diffusi approcci basati sui Vision-Language Models (VLM). Tali modelli elaborano la pagina come un'informazione multimodale, combinando le caratteristiche visuali del documento con il contenuto linguistico. Possono quindi interpretare la disposizione spaziale degli elementi e produrre rappresentazioni testuali o strutturate senza dipendere esclusivamente da una sequenza rigida di operazioni separate di OCR e analisi del layout [12].

Gli approcci basati su VLM possono risultare particolarmente adatti all'elaborazione di documenti contenenti layout complessi, tabelle, formule, immagini annotate e relazioni spaziali. L'elaborazione congiunta degli aspetti visuali e testuali può inoltre ridurre alcuni degli errori derivanti dalla separazione tra riconoscimento dei caratteri e ricostruzione della struttura. Tali approcci presentano tuttavia costi computazionali generalmente superiori e possono produrre risultati variabili in funzione della risoluzione delle pagine, della densità del testo, delle caratteristiche grafiche dei documenti e del modello utilizzato.

Dopo l'estrazione, i contenuti vengono sottoposti a una fase di preprocessing. Questa può comprendere operazioni quali la normalizzazione del testo, la ricomposizione delle parole interrotte a fine riga, la rimozione di intestazioni e piè di pagina ripetuti e il filtraggio degli elementi non rilevanti. È inoltre possibile ricostruire la gerarchia del documento attraverso il riconoscimento di titoli e sottotitoli, associando ciascun blocco testuale al relativo contesto strutturale.

Il contenuto elaborato viene infine suddiviso in porzioni più piccole, denominate *chunk*, che costituiscono le unità utilizzate durante l'indicizzazione e il retrieval. Il processo di suddivisione, indicato come *chunking*, può essere eseguito sulla base di una lunghezza prestabilita oppure rispettando i confini logici di paragrafi, sezioni e procedure. A ciascun chunk possono inoltre essere associati metadati, quali il nome del documento, il titolo della sezione, le pagine di provenienza e le relazioni con eventuali blocchi circostanti.

Le tecniche di elaborazione documentale influiscono quindi direttamente sulla qualità dell'intero sistema RAG. Errori nell'estrazione, nella ricostruzione dell'ordine di lettura o nella segmentazione possono produrre chunk incompleti, rumorosi o semanticamente incoerenti, compromettendo sia il recupero delle informazioni sia la successiva generazione della risposta.

2.4 Information Retrieval e database vettoriali

L'Information Retrieval comprende l'insieme delle tecniche impiegate per individuare, all'interno di una collezione documentale, i contenuti maggiormente rilevanti rispetto a una richiesta formulata dall'utente. In una pipeline RAG, questa fase ha il compito di selezionare le porzioni di documento da fornire al modello linguistico come contesto per la generazione della risposta.

Affinché i contenuti possano essere confrontati con una query, è necessario rappresentarli in una forma elaborabile dal sistema di ricerca. Una delle rappresentazioni utilizzate nella ricerca semantica è costituita dagli embedding, ossia vettori numerici densi che codificano caratteristiche apprese di parole, frasi o porzioni di testo. Se il modello di embedding è stato addestrato in modo adeguato rispetto al dominio e al compito considerati, testi semanticamente affini

tendono a essere rappresentati da vettori vicini nello spazio vettoriale, anche quando non condividono esattamente gli stessi termini.

Nella ricerca semantica, o *dense retrieval*, sia la query sia i contenuti documentali vengono trasformati in vettori densi mediante un modello di embedding. La rilevanza viene quindi stimata attraverso una funzione di similarità o distanza, come la similarità coseno, il prodotto scalare o la distanza euclidea. I contenuti caratterizzati dal punteggio di similarità più elevato, o dalla distanza minore, vengono considerati i candidati maggiormente pertinenti [13].

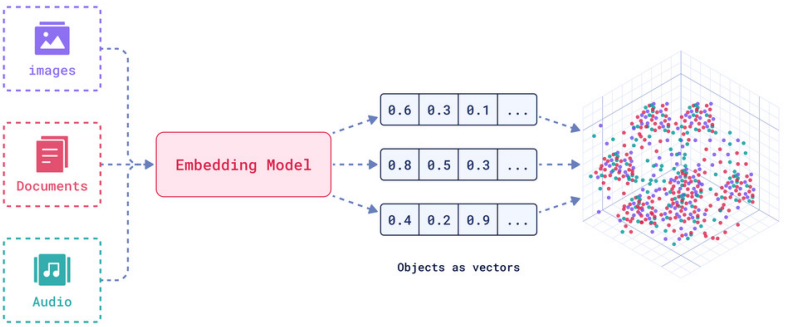


Figura 2.3: Esempio di rappresentazione densa di una query e di un documento. Il contenuto testuale viene codificato in vettori numerici compatti, nei quali ogni dimensione contribuisce alla rappresentazione semantica complessiva. Fonte: [14]

Il dense retrieval consente di recuperare contenuti concettualmente collegati alla richiesta anche quando la formulazione utilizzata dall'utente differisce da quella presente nei documenti. Tale proprietà è particolarmente utile nelle interazioni in linguaggio naturale, nelle quali uno stesso concetto può essere espresso mediante sinonimi, parafrasi o formulazioni meno tecniche [15].

Un approccio differente è rappresentato dalla ricerca lessicale, tradizionalmente basata sulla corrispondenza tra i termini presenti nella query e quelli contenuti nei documenti. Nelle rappresentazioni sparse classiche, ciascuna dimensione può essere associata a un termine del vocabolario e la maggior parte dei valori risulta nulla. La rilevanza dipende quindi dalla corrispondenza lessicale e dal peso attribuito ai termini condivisi.

Tra i metodi lessicali maggiormente utilizzati rientra BM25, una funzione di ranking che considera la frequenza dei termini nel documento, la loro rarità all'interno della collezione e la lunghezza del testo. I termini presenti in pochi documenti ricevono generalmente un peso maggiore, mentre l'aumento della frequenza di un termine all'interno dello stesso documento produce un contributo progressivamente meno rilevante al punteggio complessivo [7].

La ricerca lessicale risulta particolarmente efficace quando la query contiene denominazioni esatte di componenti, codici identificativi, sigle o termini tecnici presenti nella documentazione. Può tuttavia incontrare difficoltà quando l'utente utilizza un lessico differente da quello adottato nel documento. Il dense retrieval gestisce generalmente meglio la variabilità linguistica, ma può attribui-

re un'elevata rilevanza a contenuti semanticamente affini che non contengono il termine tecnico, il valore o il codice specificamente richiesto.

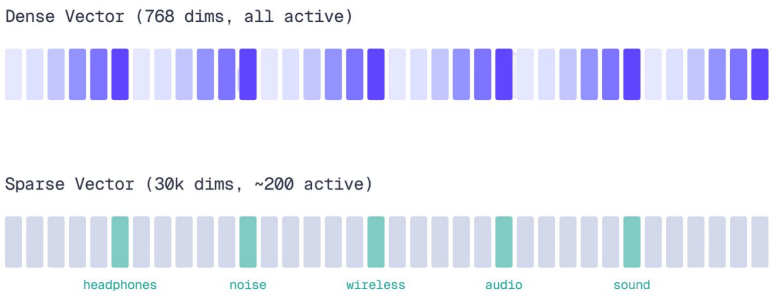


Figura 2.4: Confronto tra rappresentazione densa e rappresentazione sparsa. Nel caso denso, il testo è codificato in un vettore compatto orientato alla similarità semantica; nel caso sparso, ogni dimensione è associata a un termine del vocabolario e la maggior parte dei valori è nulla. Fonte: [16]

Poiché i due approcci presentano caratteristiche complementari, vengono spesso adottate strategie di ricerca ibrida. In tale configurazione, la query viene utilizzata per eseguire sia una ricerca densa sia una ricerca lessicale. I risultati prodotti dai diversi sistemi vengono successivamente combinati al fine di ottenere un'unica graduatoria.

La fusione può essere realizzata normalizzando e combinando i punteggi oppure utilizzando le posizioni occupate dai risultati nelle diverse graduatorie. Un metodo appartenente alla seconda categoria è la Reciprocal Rank Fusion (RRF), che assegna a ciascun risultato un punteggio calcolato sulla base dei ranghi ottenuti nei diversi sistemi di ricerca. Tale metodo consente di combinare graduatorie caratterizzate da scale di punteggio differenti senza richiedere una calibrazione diretta dei valori restituiti dai singoli retriever [17].

L'indicizzazione e la ricerca degli embedding possono essere gestite mediante un database vettoriale. Tale sistema consente di memorizzare i vettori associati ai contenuti e di eseguire efficientemente ricerche per similarità. All'aumentare del numero di elementi indicizzati, il confronto esaustivo tra il vettore della query e tutti i vettori presenti nella collezione può diventare computazionalmente oneroso. Per questo motivo vengono comunemente adottati algoritmi di Approximate Nearest Neighbor (ANN), che riducono il costo della ricerca accettando un possibile compromesso tra accuratezza del recupero e prestazioni computazionali [18].

Oltre al vettore, ciascun elemento indicizzato può includere il testo originale e un insieme di metadati, quali il nome del documento, il titolo della sezione, il numero di pagina e la tipologia del contenuto. Queste informazioni consentono di applicare filtri durante la ricerca, ricostruire il contesto dei risultati e mantenere il collegamento con le fonti originali. A seconda del sistema utilizzato, il

database può inoltre supportare congiuntamente indici densi, rappresentazioni sparse e condizioni di filtraggio sui metadati.

Nel contesto della documentazione tecnica, la combinazione tra ricerca densa e lessicale consente di gestire sia domande formulate in linguaggio naturale sia richieste contenenti termini specialistici, denominazioni di componenti, valori e codici identificativi. La qualità del retrieval dipende tuttavia non soltanto dal metodo di ricerca adottato, ma anche dal modello di embedding, dalla rappresentazione e segmentazione dei documenti, dalla configurazione degli indici e dai metadati associati ai contenuti.

Capitolo 3

Analisi del dominio e requisiti

3.1 Dominio applicativo

Il dominio applicativo del presente lavoro riguarda la consultazione della documentazione tecnica relativa agli apparati installati a bordo di yacht di grandi dimensioni e di alta gamma. Tali imbarcazioni integrano numerosi sistemi meccanici, elettrici, elettronici, idraulici e informatici, tra cui motori, pompe, sistemi per la produzione di acqua calda, impianti di ventilazione, sistemi idrici pressurizzati e stabilizzatori. Ciascun apparato è generalmente corredato da manuali tecnici specifici, forniti dal relativo costruttore.

La documentazione viene consultata da diverse figure professionali, quali comandanti, membri dell'equipaggio, tecnici di cantiere e responsabili della gestione della flotta, durante le attività di esercizio, manutenzione, collaudo e diagnostica dei guasti. Le richieste possono riguardare procedure operative, configurazioni degli apparati, interventi di manutenzione, condizioni di allarme, procedure di risoluzione dei problemi e precauzioni di sicurezza.

Nel caso d'uso considerato, l'utente formula una domanda in linguaggio naturale senza dover conoscere il titolo esatto della procedura né la terminologia adottata dal costruttore. Il sistema deve individuare le informazioni pertinenti all'interno della documentazione relativa allo specifico apparato e restituire un contesto che preservi, per quanto possibile, la sequenza delle operazioni, le condizioni preliminari, le verifiche previste e le avvertenze associate.

Il corpus documentale considerato è costituito da otto manuali tecnici in lingua inglese e in formato PDF, relativi a differenti categorie di componenti installati a bordo. Il dominio applicativo è quindi caratterizzato dall'interazione tra le esigenze informative degli utenti, la varietà delle richieste operative e la conoscenza specialistica contenuta nella manualistica tecnica.

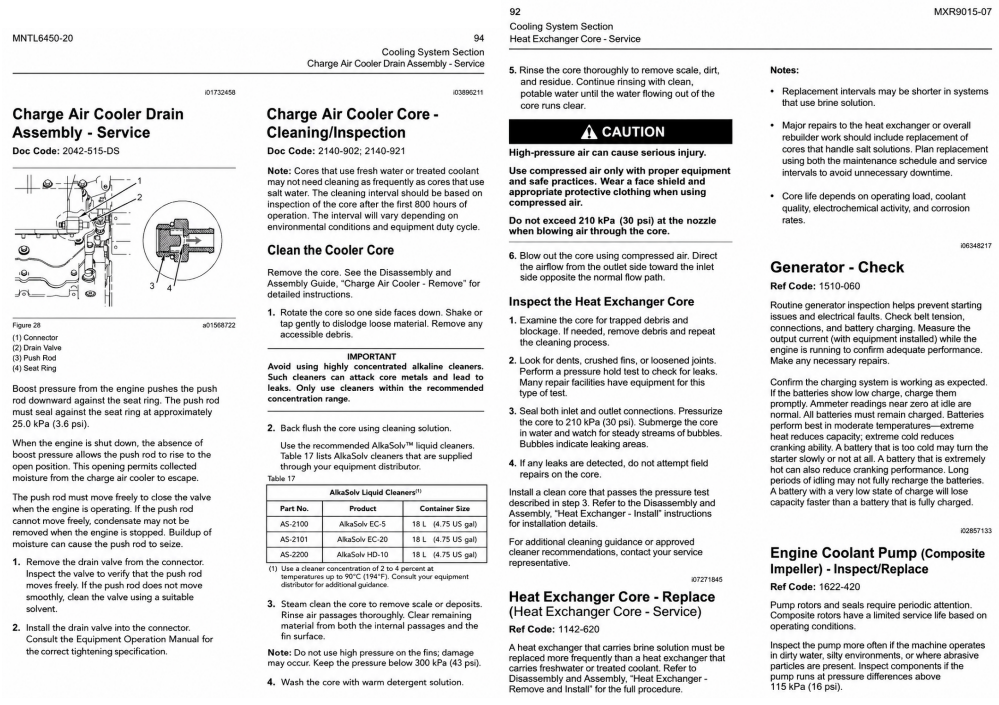
3.2 Caratteristiche dei documenti analizzati

Il corpus utilizzato nel progetto è costituito da manuali tecnici in formato PDF relativi ad apparati impiegati nel settore degli yacht di lusso. Si tratta di documentazione operativa destinata a supportare attività di consultazione, esercizio, manutenzione e gestione tecnica di sistemi complessi. I manuali non presenta-

no pertanto una natura esclusivamente descrittiva, ma comprendono procedure, avvertenze di sicurezza, note operative, riferimenti a componenti e informazioni distribuite tra più sezioni.

Una caratteristica rilevante di tali documenti è la forte dipendenza del significato dalla struttura. Le informazioni utili non sono sempre contenute in un singolo paragrafo isolato, ma possono appartenere a una procedura più ampia, composta da un titolo, una descrizione introduttiva, eventuali prerequisiti, una sequenza di operazioni, note e avvertenze. Di conseguenza, per rispondere adeguatamente a una domanda dell'utente non è sufficiente individuare una frase semanticamente affine alla query: è necessario recuperare un contesto sufficientemente ampio da consentire la corretta interpretazione del passaggio individuato.

I manuali analizzati sono redatti in lingua inglese e adottano una terminologia tecnica specifica del dominio. In molti casi, la corrispondenza lessicale tra la domanda e il manuale può risultare rilevante quanto la similarità semantica. Il nome di un componente o di una procedura, ad esempio, può comparire nel documento in una forma precisa, che deve essere preservata durante le operazioni di parsing, preprocessing, chunking e indicizzazione.



MNTL6450-20 94
Cooling System Section
Charge Air Cooler Drain Assembly - Service

i01732458

Charge Air Cooler Drain Assembly - Service

Doc Code: 2042-515-DS

Figure 28 a01568722
(1) Connector
(2) Drain Valve
(3) Push Rod
(4) Seat Ring

Boost pressure from the engine pushes the push rod downward against the seat ring. The push rod must seal against the seat ring at approximately 25.0 kPa (3.6 psi).

When the engine is shut down, the absence of boost pressure allows the push rod to rise to the open position. This opening permits collected moisture from the charge air cooler to escape.

The push rod must move freely to close the valve when the engine is operating. If the push rod cannot move freely, condensate may not be removed when the engine is stopped. Buildup of moisture can cause the push rod to seize.

1. Remove the drain valve from the connector. Inspect the valve to verify that the push rod moves freely. If the push rod does not move smoothly, clean the valve using a suitable solvent.
2. Install the drain valve into the connector. Consult the Equipment Operation Manual for the correct tightening specification.

i03896211

Charge Air Cooler Core - Cleaning/Inspection

Doc Code: 2140-902; 2140-921

Note: Cores that use fresh water or treated coolant may not need cleaning as frequently as cores that use salt water. The cleaning interval should be based on inspection of the core after the first 800 hours of operation. The interval will vary depending on environmental conditions and equipment duty cycle.

Clean the Cooler Core

Remove the core. See the Disassembly and Assembly Guide, "Charge Air Cooler - Remove" for detailed instructions.

1. Rotate the core so one side faces down. Shake or tap gently to dislodge loose material. Remove any accessible debris.

IMPORTANT
Avoid using highly concentrated alkaline cleaners. Such cleaners can attack core metals and lead to leaks. Only use cleaners within the recommended concentration range.

2. Back flush the core using cleaning solution.

   Use the recommended AlkaSolv™ liquid cleaners. Table 17 lists AlkaSolv cleaners that are supplied through your equipment distributor.

Table 17

AlkaSolv Liquid Cleaners[1]		
Part No.	**Product**	**Container Size**
AS-2100	AlkaSolv EC-5	18 L (4.75 US gal)
AS-2101	AlkaSolv EC-20	18 L (4.75 US gal)
AS-2200	AlkaSolv HD-10	18 L (4.75 US gal)

(1) Use a cleaner concentration of 2 to 4 percent at temperatures up to 90°C (194°F). Consult your equipment distributor for additional guidance.

3. Steam clean the core to remove scale or deposits. Rinse air passages thoroughly. Clear remaining material from both the internal passages and the fin surface.

Note: Do not use high pressure on the fins; damage may occur. Keep the pressure below 300 kPa (43 psi).

4. Wash the core with warm detergent solution.

92 MXR9015-07
Cooling System Section
Heat Exchanger Core - Service

5. Rinse the core thoroughly to remove scale, dirt, and residue. Continue rinsing with clean, potable water until the water flowing out of the core runs clear.

⚠ CAUTION

High-pressure air can cause serious injury.

Use compressed air only with proper equipment and safe practices. Wear a face shield and appropriate protective clothing when using compressed air.

Do not exceed 210 kPa (30 psi) at the nozzle when blowing air through the core.

6. Blow out the core using compressed air. Direct the airflow from the outlet side toward the inlet side opposite the normal flow path.

Inspect the Heat Exchanger Core

1. Examine the core for trapped debris and blockage. If needed, remove debris and repeat the cleaning process.
2. Look for dents, crushed fins, or loosened joints. Perform a pressure hold test to check for leaks. Many repair facilities have equipment for this type of test.
3. Seal both inlet and outlet connections. Pressurize the core to 210 kPa (30 psi). Submerge the core in water and watch for steady streams of bubbles. Bubbles indicate leaking areas.
4. If any leaks are detected, do not attempt field repairs on the core.

Install a clean core that passes the pressure test described in step 3. Refer to the Disassembly and Assembly, "Heat Exchanger - Install" instructions for installation details.

For additional cleaning guidance or approved cleaner recommendations, contact your service representative.

i07271845

Heat Exchanger Core - Replace (Heat Exchanger Core - Service)

Ref Code: 1142-620

A heat exchanger that carries brine solution must be replaced more frequently than a heat exchanger that carries freshwater or treated coolant. Refer to Disassembly and Assembly, "Heat Exchanger - Remove and Install" for the full procedure.

Notes:

- Replacement intervals may be shorter in systems that use brine solution.
- Major repairs to the heat exchanger or overall rebuilder work should include replacement of cores that handle salt solutions. Plan replacement using both the maintenance schedule and service intervals to avoid unnecessary downtime.
- Core life depends on operating load, coolant quality, electrochemical activity, and corrosion rates.

i06348217

Generator - Check

Ref Code: 1510-060

Routine generator inspection helps prevent starting issues and electrical faults. Check belt tension, connections, and battery charging. Measure the output current (with equipment installed) while the engine is running to confirm adequate performance. Make any necessary repairs.

Confirm the charging system is working as expected. If the batteries show low charge, charge them promptly. Ammeter readings near zero at idle are normal. All batteries must remain charged. Batteries perform best in moderate temperatures—extreme heat reduces capacity; extreme cold reduces cranking ability. A battery that is too cold may turn the starter slowly or not at all. A battery that is extremely hot can also reduce cranking performance. Long periods of idling may not fully recharge the batteries. A battery with a very low state of charge will lose capacity faster than a battery that is fully charged.

i02857133

Engine Coolant Pump (Composite Impeller) - Inspect/Replace

Ref Code: 1622-420

Pump rotors and seals require periodic attention. Composite rotors have a limited service life based on operating conditions.

Inspect the pump more often if the machine operates in dirty water, silty environments, or where abrasive particles are present. Inspect components if the pump runs at pressure differences above 115 kPa (16 psi).

Figura 3.1: Esempio di due pagine consecutive tratte da un manuale tecnico in formato PDF.

Dal punto di vista strutturale, i PDF comprendono elementi eterogenei, quali titoli di sezione e sottosezione, paragrafi, elenchi numerati o puntati, note, avvertenze, riferimenti a figure, numeri di pagina, intestazioni e piè di pagina. Tale eterogeneità rende complessa l'estrazione automatica dei contenuti, poiché il formato PDF non espone necessariamente una struttura logica esplicita. Ele-

menti graficamente simili possono infatti svolgere funzioni differenti: un testo in grassetto può rappresentare un titolo, ma anche un'etichetta, un'avvertenza o un riferimento interno. Analogamente, testi brevi o isolati possono essere classificati erroneamente come intestazioni di nuove sezioni.

Un'ulteriore difficoltà riguarda la distribuzione delle informazioni su più pagine. Alcune procedure possono iniziare in una pagina e proseguire in quella successiva oppure contenere riferimenti a informazioni collocate in altre parti del manuale. È quindi necessario conservare metadati relativi alla provenienza dei contenuti, quali il nome del documento, la sezione di appartenenza e l'intervallo di pagine. Tali informazioni risultano utili sia durante il retrieval sia nella fase di generazione della risposta, poiché consentono di fornire riferimenti verificabili e di mantenere la tracciabilità rispetto al documento originale.

La granularità della rappresentazione documentale costituisce un altro aspetto critico. Se una procedura viene mantenuta all'interno di un unico blocco molto esteso, il contesto viene preservato, ma la presenza di contenuti non direttamente pertinenti può ridurre la precisione del retrieval. Se, al contrario, il testo viene suddiviso in chunk eccessivamente piccoli, il sistema può recuperare passaggi rilevanti ma privi delle condizioni preliminari, delle avvertenze o delle operazioni circostanti necessarie alla loro corretta interpretazione. Il bilanciamento tra specificità del recupero e completezza del contesto rappresenta pertanto una delle principali motivazioni per l'adozione di strategie di segmentazione più articolate, tra cui l'approccio gerarchico parent/child.

I documenti considerati richiedono quindi modalità di elaborazione differenti rispetto a testi caratterizzati da una struttura semplice e prevalentemente lineare. Il loro valore informativo dipende non soltanto dal contenuto testuale, ma anche dalle relazioni tra gli elementi della pagina, dalla gerarchia delle sezioni e dalla continuità delle procedure. La fase di preprocessing deve pertanto distinguere i contenuti informativi dagli elementi ricorrenti del layout e dai possibili falsi segnali strutturali. Una pipeline adeguata deve preservare, per quanto possibile, l'integrità delle procedure, filtrare gli elementi non rilevanti e associare a ciascuna porzione di testo i metadati necessari al recupero e alla verifica delle fonti.

3.3 Requisiti del sistema

Il sistema sviluppato deve consentire l'interrogazione di manuali tecnici in formato PDF mediante domande formulate in linguaggio naturale. L'obiettivo principale consiste nell'individuare le procedure pertinenti all'interno della documentazione e nell'utilizzare i contenuti recuperati come contesto per la generazione di risposte tecniche, coerenti e verificabili.

Dal punto di vista funzionale, il sistema deve innanzitutto supportare una fase offline di acquisizione ed elaborazione dei documenti. Durante tale fase, i manuali devono essere analizzati e i relativi contenuti estratti, normalizzati e suddivisi in unità informative coerenti. La pipeline deve preservare, per quanto possibile, la struttura logica dei documenti, riconoscendo elementi quali titoli, sezioni, procedure ed elenchi e filtrando i contenuti non rilevanti ai fini della

ricerca. Ciascuna unità indicizzata deve inoltre essere associata a metadati quali il nome del manuale, il titolo della sezione, le pagine di provenienza e gli identificativi necessari a ricostruire le relazioni tra frammenti granulari e contesti documentali più ampi.

Un requisito centrale riguarda la qualità del retrieval. Il sistema deve essere in grado di recuperare non soltanto porzioni di testo semanticamente affini alla domanda, ma contenuti pertinenti rispetto alla procedura richiesta. A tale scopo, il processo di ricerca deve combinare segnali semantici, ottenuti mediante embedding, e segnali lessicali, particolarmente utili in presenza di termini tecnici, sigle, codici identificativi e denominazioni di componenti. La pipeline deve inoltre prevedere meccanismi per la fusione e il riordinamento dei risultati, al fine di selezionare un contesto quanto più possibile pertinente e completo da fornire al modello generativo.

La fase di generazione deve produrre risposte fondate sui contenuti recuperati e coerenti con le informazioni presenti nei manuali. Il modello deve essere istruito a non integrare la risposta con informazioni non supportate dal contesto e, qualora i contenuti disponibili risultino insufficienti, deve segnalare esplicitamente l'impossibilità di formulare una risposta adeguatamente documentata. Le risposte devono essere formulate in modo chiaro, tecnico e conciso e devono includere riferimenti alle sezioni o alle pagine dalle quali derivano le informazioni utilizzate.

Tra i requisiti non funzionali assume particolare importanza la tracciabilità. Ogni risposta deve poter essere ricondotta ai documenti originali, in modo da consentire all'utente di verificarne il contenuto. Il sistema deve inoltre essere progettato secondo un'architettura modulare, così da permettere la sostituzione e il confronto di differenti strategie di parsing, preprocessing, chunking, indicizzazione, retrieval e generazione senza modificare l'intera pipeline.

Un ulteriore requisito riguarda la riproducibilità della valutazione sperimentale. Il sistema deve consentire l'esecuzione controllata delle diverse configurazioni e la registrazione dei risultati prodotti nelle fasi di retrieval e generazione. A tale scopo è necessario disporre di un dataset di riferimento costituito da domande associate alle procedure attese, ai relativi contenuti e alle pagine di provenienza. Ciò consente di valutare separatamente la capacità di recuperare le informazioni corrette e la qualità delle risposte generate, facilitando l'individuazione della fase della pipeline nella quale si verificano eventuali errori.

Capitolo 4

Architettura della pipeline RAG

La pipeline progettata segue l'architettura generale di un sistema di Retrieval-Augmented Generation (RAG) ed è articolata in tre macrofasi: ingestion, retrieval e generazione. Ciascuna fase svolge una funzione specifica nel processo che conduce dai manuali tecnici originali alla produzione di risposte fondate sulla documentazione disponibile.

La fase di ingestion ha lo scopo di preparare la base documentale ed è eseguita prima dell'interazione con l'utente. I manuali tecnici in formato PDF vengono acquisiti ed elaborati al fine di estrarne i contenuti e ricostruirne, per quanto possibile, la struttura logica. Il testo viene successivamente normalizzato, filtrato e suddiviso in unità informative *(chunk)*, ciascuna delle quali viene arricchita con metadati relativi al documento di origine, alla sezione di appartenenza e alle pagine di provenienza.

Le unità documentali prodotte vengono quindi trasformate nelle rappresentazioni necessarie alle diverse modalità di ricerca e memorizzate all'interno del sistema di indicizzazione. L'indice risultante consente di eseguire ricerche basate sia sulla similarità semantica sia sulla corrispondenza lessicale. Poiché queste operazioni non dipendono dalle singole richieste degli utenti, il processo viene eseguito offline e ripetuto quando i documenti vengono aggiunti, modificati o rimossi.

Il flusso online ha inizio con la formulazione di una domanda in linguaggio naturale. Durante la fase di retrieval, la richiesta viene utilizzata per interrogare gli indici e individuare i contenuti maggiormente pertinenti. I risultati prodotti dalle diverse modalità di ricerca vengono successivamente combinati, filtrati e riordinati, con l'obiettivo di selezionare il contesto più rilevante rispetto alla domanda.

La pipeline adotta inoltre una rappresentazione gerarchica di tipo parent/-child. La ricerca viene eseguita su frammenti documentali granulari, denominati child, che consentono di individuare porzioni specifiche rispetto alla richiesta. Una volta selezionati i risultati, il sistema risale alle corrispondenti unità parent, caratterizzate da un contenuto più ampio. In questo modo, il retrieval può operare su unità sufficientemente specifiche, mentre il modello generativo riceve un contesto maggiormente completo, comprendente le informazioni necessarie alla corretta interpretazione della procedura.

La fase di generazione costituisce l'ultimo passaggio della pipeline. La domanda dell'utente, le istruzioni operative e il contesto recuperato vengono organizzati all'interno del prompt fornito al modello linguistico. Il modello viene istruito a formulare la risposta sulla base delle informazioni presenti nel contesto e a segnalare esplicitamente i casi in cui la documentazione recuperata non risulti sufficiente a fornire una risposta adeguatamente supportata.

La risposta finale viene presentata insieme ai riferimenti al manuale, alla sezione e alle pagine di provenienza dei contenuti utilizzati. L'architettura mantiene pertanto una separazione tra preparazione della documentazione, recupero delle informazioni e generazione della risposta. Tale modularità consente di modificare e valutare separatamente i componenti delle diverse fasi, facilitando il confronto tra le strategie analizzate nel corso del lavoro.

La Figura 4.1 mostra una rappresentazione complessiva dell'architettura e dei principali flussi di elaborazione.

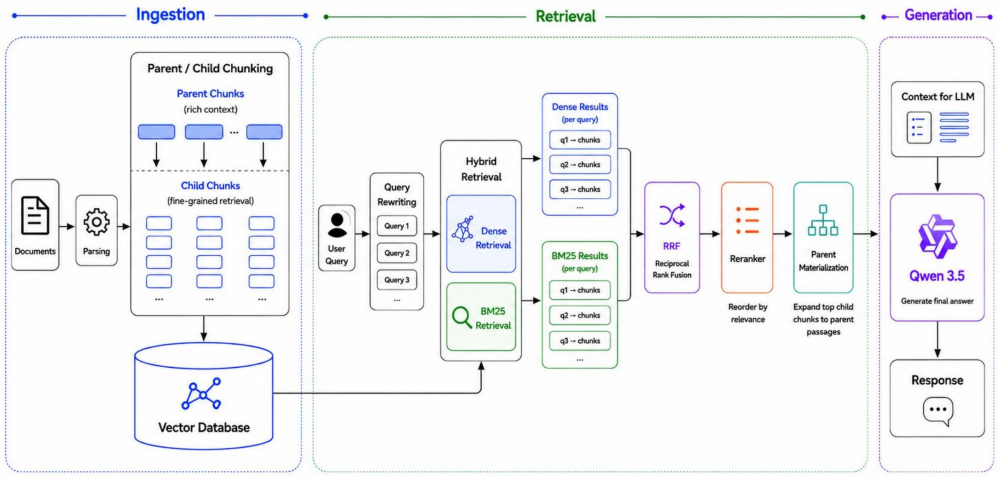


Figura 4.1: Architettura generale della pipeline RAG e distinzione tra il flusso di ingestion offline e il flusso di interrogazione online.

4.1 Flusso di ingestion offline

La fase di ingestion offline ha il compito di trasformare i manuali tecnici originali in una rappresentazione strutturata e interrogabile. Essa viene eseguita prima dell'interazione con l'utente e deve essere ripetuta, integralmente o limitatamente ai contenuti interessati, quando un documento viene aggiunto, modificato, sostituito o rimosso dalla collezione.

Il flusso ha inizio con l'acquisizione dei manuali in formato PDF. Ciascun documento viene elaborato mediante una specifica strategia di parsing, con l'obiettivo di estrarne i contenuti e ricostruirne, per quanto possibile, la struttura logica. A seconda dell'approccio adottato, l'elaborazione può basarsi sugli elementi testuali e sulle relative proprietà grafiche presenti nel PDF, su librerie specializzate nell'analisi documentale oppure su modelli capaci di considerare congiuntamente il contenuto testuale e il layout della pagina.

I contenuti estratti vengono successivamente sottoposti a preprocessing. Tale fase può comprendere la normalizzazione del testo, la ricomposizione delle parole interrotte, la rimozione di elementi ricorrenti o non informativi e il riconoscimento delle principali unità strutturali del documento, quali capitoli, sezioni, titoli, paragrafi ed elenchi. Lo scopo è ridurre il rumore introdotto dall'impaginazione e ottenere una rappresentazione quanto più possibile coerente con l'organizzazione logica del manuale.

Il contenuto elaborato viene quindi suddiviso in chunk, che costituiscono le unità impiegate durante l'indicizzazione e il retrieval. La strategia di segmentazione deve bilanciare due esigenze contrapposte: produrre porzioni sufficientemente specifiche da favorire l'individuazione dei passaggi pertinenti e preservare un contesto abbastanza ampio da mantenere comprensibili procedure, condizioni preliminari e avvertenze. Le differenti strategie di parsing, preprocessing e chunking considerate nel presente lavoro sono descritte e confrontate nel Capitolo 5.

A ciascun chunk vengono associati metadati utili a conservarne la provenienza e la collocazione all'interno del documento. Tra questi rientrano il nome del manuale, il titolo della sezione, le pagine di provenienza e gli identificativi necessari a rappresentare eventuali relazioni gerarchiche tra unità documentali. I metadati permettono inoltre di mantenere la tracciabilità dei contenuti e di riportare nella risposta finale riferimenti verificabili alla fonte originale.

Una volta prodotte le unità documentali, il loro contenuto viene trasformato mediante un modello di embedding in rappresentazioni vettoriali dense, utilizzate per la ricerca semantica. Parallelamente, vengono generate o indicizzate le rappresentazioni necessarie alla ricerca lessicale, particolarmente utile in presenza di termini tecnici, sigle, codici identificativi e denominazioni specifiche dei componenti.

L'ultima fase consiste nella memorizzazione delle unità all'interno del sistema di indicizzazione. Per ciascun elemento vengono conservati il contenuto testuale, i metadati e le rappresentazioni richieste dalle diverse modalità di ricerca. Al termine del processo di ingestion, i manuali risultano organizzati in una collezione interrogabile, che costituisce la base documentale utilizzata dal flusso di retrieval online.

4.2 Flusso di retrieval online

La fase di retrieval online ha inizio quando l'utente formula una domanda in linguaggio naturale. Il suo obiettivo è individuare, all'interno della base documentale precedentemente indicizzata, i contenuti maggiormente pertinenti e costruire il contesto da fornire al modello generativo.

La richiesta può essere sottoposta a una fase preliminare di *query rewriting*, durante la quale vengono generate una o più formulazioni alternative della domanda. Tali formulazioni devono preservare l'intento informativo originale, esprimendolo mediante termini potenzialmente più vicini al linguaggio tecnico dei manuali o maggiormente adatti alla ricerca lessicale. La query originale vie-

ne comunque mantenuta, così da limitare il rischio che la riformulazione alteri o restringa impropriamente la richiesta dell'utente.

La query originale e le eventuali formulazioni alternative vengono quindi utilizzate per interrogare il sistema di indicizzazione. La pipeline combina due modalità di ricerca complementari: il retrieval semantico, basato sul confronto tra embedding densi, e il retrieval lessicale, basato sulla corrispondenza tra i termini della richiesta e quelli presenti nei documenti. Il primo consente di recuperare contenuti espressi mediante formulazioni differenti ma semanticamente affini; il secondo risulta particolarmente efficace in presenza di codici identificativi, sigle, valori e denominazioni tecniche.

Le graduatorie prodotte dalle diverse ricerche vengono successivamente aggregate mediante un meccanismo di fusione. Tale operazione consente di integrare i risultati provenienti dai diversi retriever e di valorizzare i contenuti che occupano posizioni rilevanti in più graduatorie, evitando di dipendere esclusivamente da una singola modalità di ricerca. In questa fase viene selezionato un insieme di candidati più ampio rispetto al numero di contenuti che saranno effettivamente forniti al modello generativo.

I candidati così ottenuti vengono sottoposti a una fase di reranking. Un modello dedicato valuta più approfonditamente la relazione tra la domanda e ciascun contenuto candidato, producendo una nuova graduatoria nella quale i passaggi ritenuti maggiormente pertinenti vengono collocati nelle prime posizioni. Poiché questa operazione presenta generalmente un costo computazionale superiore rispetto alla ricerca iniziale, viene applicata soltanto a un insieme ristretto di risultati già selezionati.

Successivamente, i risultati granulari vengono utilizzati per ricostruire il contesto documentale destinato alla generazione. Quando un risultato corrisponde a un'unità child, il sistema recupera la relativa unità parent, caratterizzata da un contenuto più ampio e maggiormente contestualizzato. Questa fase, indicata come materializzazione, consente di eseguire la ricerca su frammenti specifici e, al tempo stesso, di fornire al modello informazioni sufficienti a interpretare correttamente procedure, condizioni preliminari e avvertenze.

Più risultati child possono riferirsi alla stessa unità parent. In tali casi, le unità materializzate vengono deduplicate, evitando di inserire più volte nel contesto la medesima sezione e di occupare inutilmente lo spazio disponibile nel prompt. L'ordine finale può essere determinato sulla base dei punteggi ottenuti dai frammenti associati e degli eventuali criteri adottati per preservare la coerenza documentale.

Al termine del retrieval vengono selezionati i contenuti meglio classificati, insieme ai relativi metadati, quali il titolo della sezione, il documento di origine e l'intervallo di pagine. L'insieme di tali elementi costituisce il contesto documentale utilizzato nella successiva fase di generazione della risposta.

4.3 Fase di generazione della risposta

La fase di generazione costituisce l'ultimo passaggio della pipeline RAG. Essa riceve in ingresso la domanda formulata dall'utente e i contenuti selezionati

durante il retrieval, utilizzandoli per costruire il prompt destinato al modello linguistico.

Il prompt comprende la richiesta dell'utente, le istruzioni che definiscono il comportamento atteso del modello e il contesto documentale recuperato. Quest'ultimo è costituito dai passaggi selezionati e dai relativi metadati, tra cui il nome del manuale, il titolo della sezione e le pagine di provenienza. Tali informazioni consentono di mantenere un collegamento esplicito tra la risposta generata e la documentazione originale.

Il modello viene istruito a formulare la risposta sulla base dei contenuti forniti dal sistema, evitando di introdurre valori, passaggi operativi o ipotesi non supportati dalla documentazione recuperata. Qualora il contesto non contenga informazioni sufficienti, il modello deve segnalare esplicitamente l'impossibilità di produrre una risposta adeguatamente documentata, anziché integrare le informazioni mancanti mediante conoscenze non verificabili.

La struttura della risposta viene adattata alla natura della richiesta. Nel caso di una procedura, le operazioni vengono presentate mediante una sequenza ordinata, preservando, per quanto possibile, l'ordine indicato nel manuale e riportando le eventuali condizioni preliminari e avvertenze associate. Per le richieste di carattere descrittivo viene invece prodotto un testo sintetico e tecnicamente preciso. La risposta è accompagnata dai riferimenti al documento, alla sezione e alle pagine dalle quali provengono le informazioni utilizzate.

La qualità della generazione dipende direttamente dalla pertinenza, dalla completezza e dalla correttezza del contesto fornito. Anche un modello linguistico adeguato al compito può produrre una risposta incompleta o inaccurata quando il retrieval non recupera tutte le parti necessarie di una procedura oppure introduce contenuti non pertinenti. La fase di generazione non viene pertanto considerata isolatamente, ma come il risultato dell'interazione tra elaborazione documentale, indicizzazione, recupero e selezione del contesto.

Al termine del processo, il sistema restituisce una risposta sintetica e contestualizzata, corredata dai riferimenti necessari a verificarne il contenuto mediante la consultazione diretta della documentazione originale.

Capitolo 5

Strategie a confronto

Il presente capitolo descrive e confronta le strategie considerate per trasformare i manuali tecnici in rappresentazioni documentali adatte all'indicizzazione e al retrieval. L'obiettivo è analizzare come le scelte relative al parsing, al preprocessing e al chunking influenzino la struttura delle unità indicizzate e, di conseguenza, la capacità del sistema di recuperare contenuti pertinenti e sufficientemente contestualizzati.

Le soluzioni considerate comprendono la baseline aziendale, un approccio basato sulla libreria Unstructured, una pipeline personalizzata, una strategia basata su MinerU e un approccio black-box. Le sezioni successive descrivono il funzionamento di ciascuna soluzione, le modalità con cui vengono estratti e segmentati i contenuti e le principali caratteristiche che le distinguono. Il confronto quantitativo delle prestazioni ottenute dalle diverse configurazioni sarà invece presentato nei capitoli dedicati alla valutazione sperimentale e all'analisi dei risultati.

5.1 Baseline aziendale

La prima strategia considerata corrisponde alla pipeline di chunking già disponibile in azienda, indicata nel progetto come *h_ service*. Essa costituisce la baseline rispetto alla quale sono state confrontate le successive soluzioni di elaborazione documentale.

La pipeline elabora direttamente i documenti PDF mediante la libreria PyMuPDF. Per ciascuna pagina vengono analizzati i blocchi testuali, le righe e i relativi span, dai quali sono ricavati il testo, la dimensione del carattere e il nome del font. Il riconoscimento della struttura documentale si basa quindi principalmente sulle proprietà tipografiche associate agli elementi estratti.

Un elemento testuale viene interpretato come titolo principale quando la dimensione del carattere supera una prima soglia configurabile. I sottotitoli vengono invece riconosciuti mediante una soglia inferiore, combinata con la presenza di un font in grassetto o con il superamento di un'ulteriore soglia dimensionale. Quando viene individuato un nuovo titolo o sottotitolo, il testo accumulato fino a quel momento viene utilizzato per creare un chunk; i contenuti successivi vengono quindi associati alla nuova sezione o sottosezione.

La pipeline riconosce inoltre gli eventuali codici SMCS presenti nel testo mediante un'espressione regolare. Al momento della creazione del chunk, il contenuto viene preceduto da una descrizione strutturata comprendente la sezione principale, l'argomento e, quando disponibile, il codice SMCS. Gli elementi costituiti esclusivamente da cifre vengono ignorati, mentre i chunk con una lunghezza inferiore a una soglia minima vengono scartati. Gli spazi e le interruzioni presenti nel testo accumulato vengono infine normalizzati.

A ciascun chunk vengono associati un identificativo stabile, il nome del documento, la sezione principale, la sottosezione e un singolo riferimento di pagina. Quest'ultimo non rappresenta tuttavia l'intero intervallo occupato dal contenuto: nell'implementazione viene aggiornato durante la lettura del documento e corrisponde alla pagina dell'ultimo elemento testuale inserito nel buffer prima della creazione del chunk. Di conseguenza, una sezione distribuita su più pagine non conserva esplicitamente la propria pagina iniziale.

Il principale vantaggio della strategia risiede nella semplicità dell'approccio. La pipeline sfrutta direttamente le caratteristiche tipografiche dei manuali e non richiede modelli dedicati all'analisi del layout. Quando la formattazione del documento è regolare e coerente, il metodo consente di separare i contenuti in corrispondenza dei principali titoli e sottotitoli, mantenendo nello stesso chunk il testo compreso tra due intestazioni consecutive.

L'approccio presenta tuttavia alcuni limiti. Il riconoscimento della struttura dipende in larga misura dalle dimensioni e dallo stile dei caratteri. Di conseguenza, elementi graficamente evidenziati ma non corrispondenti a vere sezioni possono essere interpretati come intestazioni; al contrario, titoli formattati in modo differente rispetto alle soglie configurate possono non essere riconosciuti. Il metodo non applica inoltre regole specifiche per distinguere titoli, etichette, avvertenze, didascalie o altri elementi caratterizzati da una formattazione simile.

Un'ulteriore criticità riguarda la ricostruzione dell'ordine di lettura. Gli span vengono elaborati secondo l'ordine restituito dal parser del PDF, senza una fase esplicita di analisi delle relazioni spaziali tra colonne, tabelle e riquadri. Nei documenti caratterizzati da layout complessi, il testo accumulato può quindi non riflettere correttamente l'organizzazione visuale della pagina.

Il contenuto viene inoltre accumulato fino al riconoscimento del titolo o del sottotitolo successivo, senza imporre una dimensione massima ai chunk e senza applicare meccanismi di sovrapposizione. Le sezioni più estese possono pertanto produrre blocchi di grandi dimensioni, contenenti informazioni eterogenee e soltanto parzialmente pertinenti rispetto alla domanda. Ciò può introdurre rumore nella rappresentazione vettoriale e ridurre la precisione del retrieval.

La baseline non adotta infine una rappresentazione gerarchica parent/child: ciascun chunk viene trattato come un'unità indipendente, con il campo relativo al parent non valorizzato. La strategia rappresenta quindi un punto di partenza semplice e coerente con la struttura tipografica dei documenti, ma offre un controllo limitato sulla granularità, sulla continuità tra pagine e sull'interpretazione degli elementi grafici.

5.2 Approccio basato su Unstructured

La seconda strategia analizzata utilizza la libreria Unstructured per l'estrazione e la classificazione degli elementi presenti nei documenti PDF. Rispetto alla baseline aziendale, questo approccio non determina la struttura esclusivamente mediante soglie definite sulla dimensione e sullo stile dei caratteri, ma delega a uno strumento specializzato l'analisi del layout e l'assegnazione delle categorie documentali.

L'estrazione viene eseguita mediante la funzione `partition_pdf`, configurata con la strategia `hi_res`, con il riconoscimento della struttura delle tabelle abilitato e con la lingua inglese come riferimento. Il documento viene così trasformato in una sequenza di elementi classificati in categorie quali titoli, testo narrativo, intestazioni ed elementi di lista. Per ciascun elemento vengono recuperati il contenuto testuale, la categoria assegnata e il numero di pagina riportato nei relativi metadati.

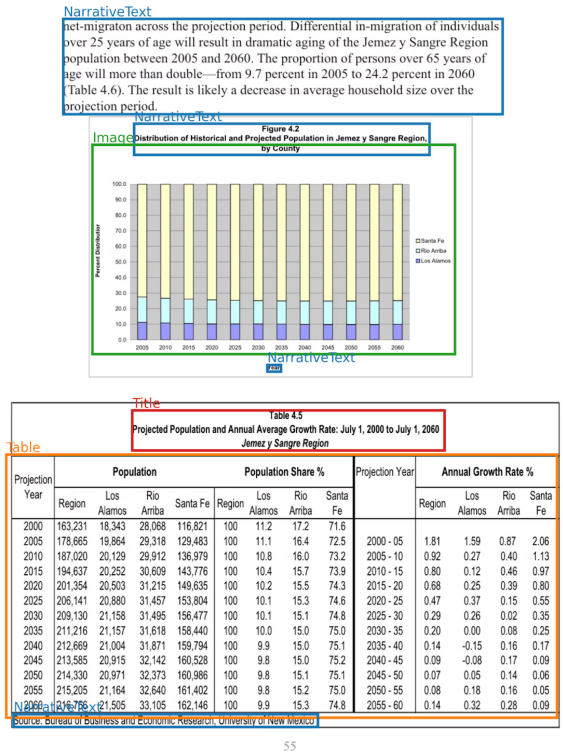


Figura 5.1: Esempio di analisi del layout prodotta da Unstructured. Fonte: [19]

Gli elementi classificati come `Title` vengono utilizzati direttamente per individuare l'inizio di una nuova sezione. Quando viene incontrato un nuovo titolo, la sezione precedente viene chiusa e gli elementi accumulati vengono associati al titolo corrente. Il testo compreso tra due titoli consecutivi costituisce quin-

di la base per la creazione di un'unità documentale parent, che rappresenta il contenuto complessivo della sezione.

All'interno di ciascuna sezione, gli elementi consecutivi classificati come `ListItem` vengono aggregati in un unico blocco. Tale scelta mira a evitare che i punti appartenenti alla stessa lista vengano separati durante la segmentazione, preservando maggiormente la continuità delle procedure numerate o puntate. Gli altri elementi vengono invece mantenuti come blocchi distinti, insieme ai rispettivi riferimenti di pagina.

La strategia adotta una rappresentazione gerarchica di tipo parent/child. Il parent contiene l'intero testo della sezione, preceduto dal relativo titolo, mentre i child vengono prodotti suddividendo i blocchi della sezione secondo una dimensione massima configurabile. Nell'implementazione utilizzata, i valori predefiniti sono pari a 1800 caratteri per ciascun child e a 150 caratteri di sovrapposizione tra frammenti consecutivi. I child identici al rispettivo parent, eventualità che può verificarsi per sezioni particolarmente brevi, non vengono creati come unità separate.

A ciascuna unità vengono associati un identificativo stabile, il titolo, il nome del documento, l'intervallo di pagine e le informazioni necessarie a rappresentare la relazione gerarchica. I child contengono infatti l'identificativo del relativo parent, mentre le unità parent sono contrassegnate esplicitamente come tali. La pipeline tenta inoltre di ricavare informazioni di sezione e capitolo dagli elementi classificati come intestazioni o titoli presenti nelle singole pagine.

Il principale vantaggio dell'approccio consiste nella disponibilità di una rappresentazione documentale già parzialmente strutturata. La classificazione automatica consente di distinguere tipologie differenti di contenuto e di applicare trattamenti specifici, come la delimitazione delle sezioni mediante i titoli e l'aggregazione degli elementi di lista. Rispetto alla baseline, la presenza di una dimensione massima per i child permette inoltre di controllare maggiormente la granularità delle unità utilizzate nel retrieval.

La qualità del risultato dipende tuttavia dall'accuratezza con cui Unstructured classifica gli elementi del PDF. Nell'implementazione considerata, ogni elemento appartenente alla categoria `Title` determina direttamente l'apertura di una nuova sezione, senza l'applicazione di ulteriori regole di validazione. Un testo classificato erroneamente come titolo può quindi produrre una frammentazione impropria, mentre un titolo non riconosciuto può causare l'unione di contenuti appartenenti a sezioni differenti.

Un'ulteriore criticità riguarda la segmentazione dei child. Sebbene le liste vengano mantenute come blocchi unitari, la sovrapposizione viene calcolata sugli ultimi caratteri del chunk precedente e può quindi iniziare nel mezzo di una parola, di una frase o di un elemento strutturale. Inoltre, un singolo blocco di lunghezza superiore alla dimensione massima non viene ulteriormente suddiviso, poiché il limite viene applicato durante l'aggregazione dei blocchi e non all'interno di ciascun blocco.

L'approccio basato su Unstructured offre pertanto una rappresentazione più articolata e un maggiore controllo sulla granularità rispetto alla baseline aziendale, ma mantiene una dipendenza significativa dalla qualità della classificazione

automatica e dalle modalità con cui gli elementi estratti vengono trasformati in sezioni e chunk.

5.3 Approccio basato su una pipeline custom

La terza strategia analizzata consiste in una pipeline custom, sviluppata con l'obiettivo di aumentare il controllo sull'estrazione e sulla segmentazione dei manuali tecnici. Come la baseline aziendale, essa elabora direttamente i documenti PDF mediante PyMuPDF; rispetto a quest'ultima, introduce tuttavia un insieme più articolato di euristiche per il riconoscimento della struttura documentale e la costruzione dei chunk.

Durante l'estrazione, il documento viene analizzato a livello di blocchi, righe e span testuali. Per ciascuna riga vengono considerate proprietà quali la dimensione del carattere e la presenza del grassetto. Un elemento può essere classificato come titolo quando supera le soglie tipografiche configurate oppure, in determinate condizioni, quando è composto da testo in maiuscolo e grassetto. Le righe di testo adiacenti appartenenti alla stessa pagina vengono inoltre aggregate, mentre le intestazioni distribuite su più righe possono essere ricomposte quando vengono riconosciute come parti dello stesso titolo.

La pipeline applica successivamente regole specifiche per escludere elementi che, pur presentando caratteristiche grafiche simili a quelle di un titolo, non identificano l'inizio di una nuova sezione. Tra questi rientrano singole lettere, elementi di lista, intervalli di pagine, riferimenti a figure, etichette di avvertenza quali *Warning*, *Caution* e *Notice*, indicazioni di nota e testi conclusi da determinati segni di punteggiatura. Tale filtraggio mira a limitare la creazione di sezioni spurie e a preservare maggiormente la continuità delle procedure.

La posizione verticale del testo viene utilizzata per analizzare le intestazioni presenti nella parte superiore delle pagine. A partire da tali elementi, il sistema tenta di ricostruire il capitolo e la sezione di appartenenza, propagando il contesto alle pagine successive quando non vengono individuate nuove intestazioni. Queste informazioni vengono associate ai chunk insieme al titolo riconosciuto, al nome del documento e all'intervallo effettivo delle pagine di provenienza.

Il contenuto viene organizzato mediante una rappresentazione gerarchica parent/child. Ogni sezione delimitata da un titolo genera un'unità parent contenente il testo complessivo della sezione. Quando la sua estensione lo richiede, il contenuto viene inoltre suddiviso in unità child più granulari, utilizzando una dimensione massima configurabile e una sovrapposizione tra frammenti consecutivi. I child mantengono un riferimento esplicito al relativo parent, permettendo di utilizzare unità specifiche durante la ricerca e di recuperare successivamente un contesto documentale più ampio.

Il principale vantaggio dell'approccio consiste nel controllo diretto esercitato sulle diverse fasi di elaborazione. Le regole possono essere adattate alle caratteristiche ricorrenti dei manuali analizzati, consentendo di intervenire sui criteri di riconoscimento dei titoli, sul filtraggio dei falsi segnali strutturali, sulla ricostruzione dei metadati e sulla granularità dei chunk. Tale flessibilità comporta tuttavia una maggiore dipendenza da euristiche definite manualmente, il

cui comportamento può variare in presenza di documenti caratterizzati da stili tipografici e strutture differenti.

La progettazione e l'implementazione della pipeline personalizzata, comprese le strategie di parsing, filtraggio, chunking, gestione dei metadati e indicizzazione, sono descritte in dettaglio nel Capitolo 6.

5.4 Approccio basato su MinerU2.5

La quarta strategia analizzata utilizza MinerU, uno strumento open source per il parsing di documenti complessi e la loro conversione in formati elaborabili automaticamente, tra cui Markdown e JSON. MinerU è progettato per estrarre e organizzare diverse tipologie di contenuto, quali testo, immagini, tabelle e formule, ricostruendo al tempo stesso il layout e l'ordine di lettura del documento [20].

Nel presente lavoro è stato utilizzato MinerU2.5, un Vision-Language Model da 1,2 miliardi di parametri specializzato nel parsing di documenti ad alta risoluzione. Il modello adotta una strategia di elaborazione *coarse-to-fine* articolata in due fasi e, secondo i risultati riportati dagli autori, raggiunge prestazioni allo stato dell'arte su diversi benchmark relativi al riconoscimento del layout e dei contenuti documentali [21].

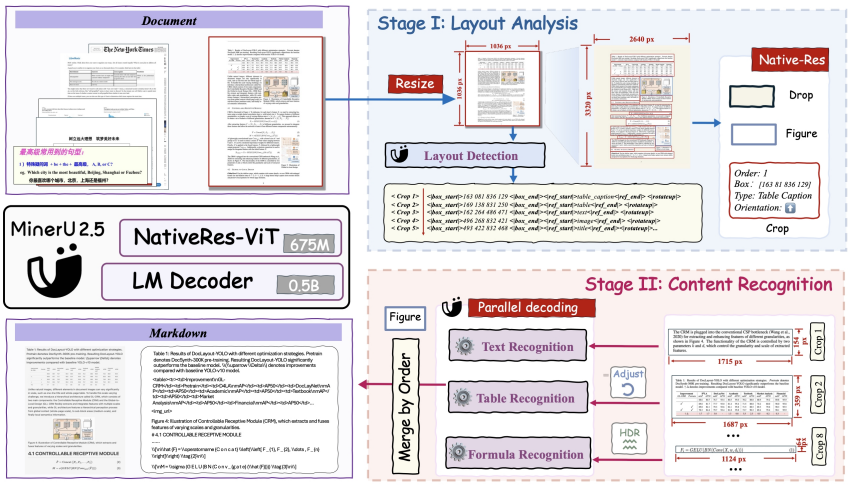


Figura 5.2: Architettura MinerU2.5. Fonte: [22]

Durante la prima fase, la pagina viene analizzata a risoluzione ridotta per individuarne la struttura globale, localizzare le principali regioni e determinarne la tipologia e l'ordine di lettura. Nella seconda fase, le regioni individuate vengono ritagliate dall'immagine originale e analizzate alla risoluzione nativa, così da preservare i dettagli presenti in aree caratterizzate da testo denso, formule o tabelle. Tale separazione consente di evitare l'elaborazione integrale della pagina ad alta risoluzione, limitando il costo computazionale associato a regioni vuote o poco informative [21].

Nel flusso sperimentale, MinerU è stato eseguito esternamente alla pipeline RAG mediante interfaccia a riga di comando:

```
mineru -p <input_path> -o <output_path>
```

Per ciascun manuale, il processo ha prodotto una directory contenente il documento convertito in Markdown e diversi file intermedi relativi al contenuto e al layout. La pipeline sviluppata non esegue pertanto direttamente il modello VLM, ma riceve e sottopone a preprocessing gli artefatti generati durante questa fase preliminare.

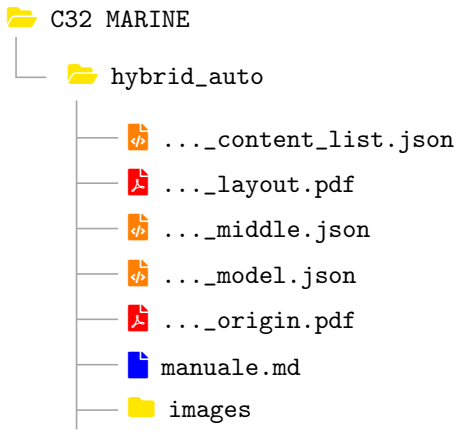


Figura 5.3: Struttura della directory prodotta da MinerU

In particolare, vengono utilizzati il file Markdown e il corrispondente file `_content_list.json`. Il primo viene analizzato riga per riga: le intestazioni Markdown, identificate dai caratteri `#`, vengono rappresentate come titoli, mentre le altre righe non vuote vengono trattate come contenuto narrativo. Il file JSON viene invece utilizzato per associare ai titoli individuati le pagine del documento originale, sfruttando gli elementi testuali classificati al primo livello della gerarchia prodotta da MinerU.

Prima della costruzione delle sezioni, il contenuto Markdown viene normalizzato e filtrato. Vengono rimossi i riferimenti alle immagini, i tag HTML, i collegamenti, i marcatori di formattazione, i frammenti di codice e le espressioni matematiche. La pipeline applica inoltre regole per evitare che elementi quali numeri isolati, riferimenti a figure, intervalli di pagine, elementi di lista ed etichette di sicurezza vengano interpretati come titoli di sezione.

Le sezioni vengono delimitate a partire dai titoli considerati validi e organizzate secondo una rappresentazione gerarchica parent/child. Ogni parent contiene il testo complessivo della sezione, mentre i child vengono generati mediante una dimensione massima configurabile e una sovrapposizione tra frammenti consecutivi. Come nelle altre pipeline gerarchiche considerate, i child vengono utilizzati come unità granulari per il retrieval, mentre i parent consentono di ricostruire un contesto più ampio durante la generazione della risposta.

Il principale vantaggio di questo approccio consiste nella possibilità di partire da una rappresentazione che incorpora già informazioni ricavate dall'analisi visuale e strutturale delle pagine, anziché tentare di ricostruire l'organizzazione del documento esclusivamente a partire dal testo interno del PDF. La conver-

sione in Markdown fornisce inoltre una rappresentazione intermedia facilmente ispezionabile e modificabile.

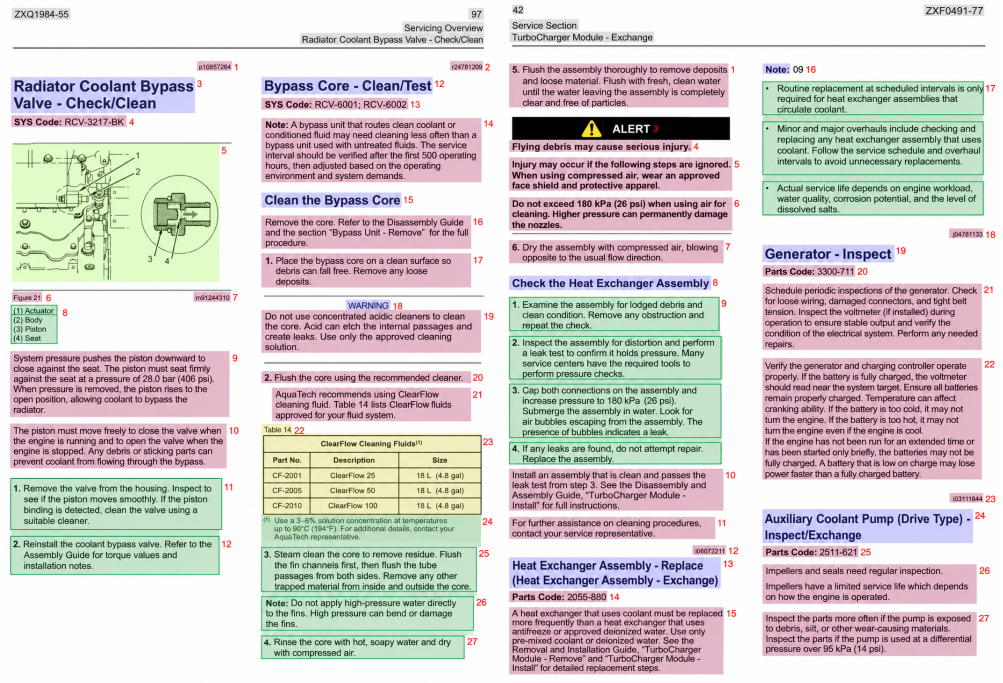


Figura 5.4: Esempi di analisi del layout prodotta da MinerU

L'approccio dipende tuttavia dalla qualità della conversione effettuata da MinerU. Eventuali errori nel riconoscimento del layout, dei titoli, del testo o dell'ordine di lettura vengono trasferiti agli artefatti prodotti e possono influenzare le successive fasi di segmentazione e retrieval. Il preprocessing implementato utilizza inoltre soltanto una parte delle informazioni disponibili: immagini e formule vengono eliminate, mentre il file JSON è impiegato principalmente per ricostruire l'associazione tra titoli e pagine.

Un'ulteriore limitazione riguarda la precisione dei metadati di pagina. Gli elementi testuali collocati dopo un titolo ereditano la pagina associata a quest'ultimo fino al riconoscimento del titolo successivo. Tale meccanismo consente di conservare una localizzazione approssimativa delle sezioni, ma non ricostruisce necessariamente la pagina esatta di ciascun paragrafo. Infine, l'esecuzione preventiva del modello introduce una fase aggiuntiva di elaborazione e requisiti computazionali superiori rispetto al parsing testuale diretto.

5.5 Approccio black-box

L'ultima strategia considerata consiste in un approccio black-box basato su un modello multimodale della famiglia Gemini. A differenza delle pipeline descritte nelle sezioni precedenti, questa configurazione non prevede fasi esplicitamente implementate di parsing, chunking, indicizzazione e retrieval. Il manuale

in formato PDF viene fornito direttamente al modello insieme alla domanda dell'utente, delegando al sistema multimodale l'individuazione della procedura pertinente e l'estrazione delle informazioni richieste.

Per ciascun manuale, il file PDF viene caricato mediante le API Gemini e mantenuto disponibile per l'elaborazione delle domande associate allo stesso documento. Prima di avviare le interrogazioni, il sistema verifica che il file abbia raggiunto lo stato attivo. Il documento caricato viene quindi riutilizzato per tutte le query appartenenti al relativo dataset e rimosso dal servizio al termine dell'elaborazione.

Per ogni domanda viene costruito un prompt di estrazione che richiede al modello di leggere direttamente il PDF e individuare la singola procedura maggiormente pertinente rispetto alla richiesta dell'utente. Il modello deve restituire il titolo completo della procedura, il relativo contenuto testuale e i numeri delle pagine nelle quali essa compare. Il prompt specifica inoltre che devono essere inclusi tutti gli elementi rilevanti, quali passaggi ordinati, avvertenze, note, condizioni preliminari e requisiti, evitando di sintetizzare il contenuto o introdurre informazioni non presenti nel documento.

Qualora la procedura richiesta non venga individuata, il modello viene istruito a restituire stringhe vuote e una lista di pagine priva di elementi. La risposta deve essere prodotta esclusivamente in formato JSON secondo uno schema predefinito composto dai campi `procedure_title`, `procedure_text` e `pages`. Il formato strutturato è richiesto anche mediante l'impostazione del tipo MIME della risposta a `application/json`.

```
{
    "procedure_title": "Radiator Coolant Bypass Valve - Check/
    Clean",
    "procedure_text": "1. Remove the valve from the housing.
    Inspect to see if the piston moves smoothly. If the piston
    binding is detected, clean the valve using a suitable
    cleaner. \n...\n",
    "pages": [87, 88]
}
```

Listing 5.1: Esempio di risposta restituita da Gemini

Dopo la ricezione della risposta, l'output viene deserializzato e normalizzato. I campi testuali mancanti vengono sostituiti con stringhe vuote, mentre i numeri di pagina vengono convertiti in valori interi e filtrati, eliminando valori non validi, duplicati o non positivi. Questa fase consente di ottenere una struttura uniforme per le successive operazioni di valutazione.

Il principale vantaggio dell'approccio risiede nella semplicità architetturale. Non è necessario progettare separatamente le fasi di estrazione, segmentazione, indicizzazione e recupero, poiché il modello riceve direttamente l'intero documento e la domanda. La natura multimodale del sistema permette inoltre di considerare congiuntamente il contenuto testuale e la struttura visuale delle pagine.

L'approccio presenta tuttavia una limitata osservabilità. Il processo me-

diante il quale il modello analizza il documento, seleziona le pagine e delimita la procedura rimane interno al servizio e non può essere esaminato o controllato separatamente. In caso di errore risulta quindi difficile stabilire se la causa sia riconducibile alla lettura del PDF, all'individuazione della sezione pertinente, alla delimitazione della procedura o alla generazione dell'output JSON.

L'assenza di un indice documentale persistente comporta inoltre che il PDF debba essere reso disponibile al modello durante l'elaborazione delle richieste. Sebbene il file venga riutilizzato per tutte le query associate allo stesso manuale, l'approccio può presentare costi e tempi di elaborazione superiori rispetto a una pipeline basata su retrieval, soprattutto in presenza di documenti estesi o di un numero elevato di interrogazioni.

Questa configurazione non costituisce propriamente una pipeline RAG, poiché non espone un componente di retrieval separato né una base documentale indicizzata. È stata pertanto impiegata come baseline multimodale end-to-end, utile per confrontare una soluzione interamente delegata al modello con le pipeline nelle quali le fasi di elaborazione documentale e recupero delle informazioni sono esplicitamente progettate e valutabili.

Capitolo 6

Progettazione e sviluppo della pipeline

Il presente capitolo descrive la progettazione e l'implementazione della pipeline personalizzata sviluppata durante il tirocinio per l'elaborazione e l'interrogazione dei manuali tecnici. La soluzione è stata realizzata con l'obiettivo di mantenere un controllo esplicito sulle diverse fasi del processo, dall'estrazione e dalla rappresentazione dei contenuti dei documenti PDF fino al recupero delle informazioni e alla generazione della risposta.

La fase offline comprende il parsing dei documenti, il filtraggio degli elementi non rilevanti, la ricostruzione della struttura dei manuali e la suddivisione gerarchica dei contenuti mediante una strategia parent/child. A ciascuna unità documentale vengono inoltre associati metadati finalizzati a conservarne la provenienza, la collocazione all'interno del manuale e le relazioni con le unità appartenenti alla stessa sezione.

I chunk prodotti vengono successivamente trasformati nelle rappresentazioni necessarie alle diverse modalità di ricerca e indicizzati in Qdrant. La base di dati risultante supporta sia il retrieval semantico, basato su embedding densi, sia il retrieval lessicale, utile per individuare termini tecnici, sigle, codici identificativi e denominazioni specifiche presenti nei manuali.

Durante la fase online, la richiesta dell'utente viene utilizzata per interrogare gli indici mediante una strategia di retrieval ibrido. I risultati provenienti dalle diverse modalità di ricerca vengono combinati e successivamente riordinati attraverso una fase di reranking. Quando i contenuti selezionati corrispondono a unità child, il sistema recupera le rispettive unità parent, così da costruire un contesto più ampio e maggiormente adatto alla corretta interpretazione delle procedure.

Il contesto così ottenuto viene infine organizzato, insieme alla domanda dell'utente e alle istruzioni operative, nel prompt fornito al modello linguistico. Le sezioni successive approfondiscono le principali scelte progettuali e implementative, con particolare attenzione al riconoscimento della struttura dei manuali, alla gestione della granularità dei chunk, alla conservazione dei metadati, all'indicizzazione, al retrieval ibrido, al reranking e alla definizione del prompt di generazione.

6.1 Parsing e filtraggio

La prima fase della pipeline personalizzata riguarda l'estrazione dei contenuti dai manuali PDF e il riconoscimento degli elementi che ne definiscono la struttura. L'obiettivo non consiste nell'ottenere una semplice sequenza lineare di testo, ma nel conservare informazioni utili a distinguere i titoli dal corpo del documento e a ricostruire, per quanto possibile, la gerarchia delle sezioni.

Il parsing viene eseguito mediante la libreria PyMuPDF. Per ciascuna pagina vengono analizzati i blocchi testuali, le righe e i singoli span che le compongono. Da ogni riga vengono ricavati il contenuto testuale, la dimensione massima del carattere, l'eventuale presenza di un font in grassetto, la posizione verticale nella pagina e gli identificativi del blocco e della riga di appartenenza. Queste informazioni vengono utilizzate sia per il riconoscimento preliminare dei titoli sia per la successiva ricostruzione del contesto documentale.

I testi degli span appartenenti alla stessa riga vengono normalizzati e concatenati, eliminando spaziature irregolari. Le righe vuote, quelle costituite da un solo carattere e quelle composte esclusivamente da cifre vengono scartate, poiché non forniscono generalmente informazioni utili alla costruzione delle sezioni. Il risultato iniziale è una sequenza ordinata di elementi associati alla relativa pagina e alle proprietà tipografiche rilevate.

L'identificazione preliminare dei titoli si basa su soglie configurabili. Una riga viene classificata come candidata quando la dimensione massima del carattere supera la soglia prevista per i titoli principali. Può inoltre essere riconosciuta come titolo quando supera la soglia dei sottotitoli ed è in grassetto oppure quando la dimensione eccede ulteriormente tale soglia. Un criterio aggiuntivo considera le righe composte da testo in maiuscolo, evidenziate in grassetto e caratterizzate da una dimensione del font superiore a un valore minimo.

Le sole proprietà tipografiche non sono tuttavia sufficienti a stabilire se una riga rappresenti l'inizio effettivo di una sezione. I manuali tecnici contengono infatti numerosi elementi graficamente evidenziati, quali avvertenze, riferimenti a figure, etichette operative e indicatori di pagina, che non devono determinare la creazione di una nuova unità documentale. Per questo motivo, i candidati vengono sottoposti a una seconda fase di validazione basata su regole testuali.

Sono esclusi gli elementi costituiti da una singola lettera, quelli che presentano una struttura assimilabile a una voce di elenco, i riferimenti alle figure e le espressioni riconducibili a intervalli di pagine. Vengono inoltre scartate le etichette associate ad avvertenze e note, tra cui *Warning*, *Caution*, *Notice*, *Danger*, *Important* e *Note*, comprese le varianti nelle quali le lettere risultano separate da spazi. Non vengono infine considerati titoli gli elementi introdotti dalla sigla *N.B.* e le righe che terminano con due punti, punto interrogativo o punto esclamativo.

La pipeline applica inoltre una regola specifica a una serie di etichette brevi e ricorrenti, quali *Fault*, *Cause*, *Solution*, *Activity*, *Action*, *On*, *Off*, *Center*, *Up*, *Down* e *Important*. Quando una riga è composta esclusivamente da una o più di queste espressioni, essa non viene utilizzata per delimitare una nuova sezione. Tale filtro è stato introdotto per ridurre il rischio che intestazioni interne a tabelle o procedure vengano interpretate come titoli strutturali.

Una particolare attenzione è dedicata alla ricomposizione degli elementi consecutivi. Le righe appartenenti al corpo del testo e presenti nella stessa pagina vengono aggregate in un unico elemento, così da ridurre la frammentazione introdotta dalla rappresentazione interna del PDF. Anche i titoli distribuiti su più righe possono essere ricomposti quando appartengono allo stesso blocco e risultano consecutivi. Come criterio di continuità aggiuntivo, una riga che inizia con una lettera minuscola può essere interpretata come prosecuzione del titolo precedente. Le etichette isolate relative ad avvertenze e note vengono invece escluse da tale ricomposizione.

L'aggregazione delle righe di corpo semplifica la successiva costruzione delle sezioni, ma comporta anche una perdita parziale della granularità originaria. Paragrafi, elementi di lista e altri blocchi distinti presenti sulla stessa pagina possono infatti essere concatenati prima del chunking. La struttura conservata dalla pipeline dipende quindi principalmente dal riconoscimento dei titoli e dai riferimenti di pagina, più che dalla separazione originaria tra i singoli blocchi testuali.

La pipeline analizza inoltre la parte superiore di ciascuna pagina per ricavare il contesto documentale. In particolare, vengono considerate come possibili intestazioni le righe collocate entro il 22% iniziale dell'altezza della pagina. Da tali elementi il sistema tenta di ricostruire il capitolo e la sezione corrente, filtrando righe costituite esclusivamente da numeri e codici del manuale riconducibili al formato previsto. Quando una pagina non presenta nuove informazioni di intestazione, il capitolo e la sezione individuati in precedenza vengono propagati alle pagine successive.

Se tra le righe candidate compare un elemento contenente il termine *Section*, questo viene interpretato come riferimento al capitolo, mentre la riga successiva viene utilizzata come nome della sezione. In assenza di tale struttura, la pipeline adotta come soluzione di ripiego le prime due righe valide dell'intestazione. Le informazioni ricostruite non determinano direttamente i confini dei chunk, ma vengono conservate come metadati associati alle sezioni prodotte.

Il risultato del parsing è un insieme ordinato di elementi, ciascuno associato al testo, alla pagina e a un indicatore che specifica se sia stato preliminarmente riconosciuto come titolo. A questo insieme si affianca una struttura contenente il contesto di capitolo e sezione ricostruito per ciascuna pagina. Tali rappresentazioni costituiscono l'ingresso della successiva fase di segmentazione gerarchica.

L'approccio adottato mantiene un controllo esplicito sui criteri di estrazione, ricomposizione e filtraggio e consente di adattare il comportamento della pipeline alle caratteristiche ricorrenti dei manuali analizzati. Trattandosi di una soluzione basata su euristiche, la sua efficacia dipende tuttavia dalla regolarità tipografica e strutturale dei documenti e può richiedere una regolazione delle soglie o delle regole in presenza di layout significativamente differenti.

6.2 Strategia parent/child

La strategia parent/child è stata adottata per bilanciare due esigenze contrapposte: aumentare la precisione del retrieval mediante unità testuali di dimensione contenuta e preservare un contesto sufficientemente ampio per la corretta interpretazione delle procedure tecniche.

La pipeline costruisce innanzitutto le sezioni del documento a partire dagli elementi riconosciuti come titoli validi. Quando viene individuato un nuovo titolo, la sezione corrente viene chiusa e il contenuto accumulato viene associato al titolo precedente. Gli elementi che precedono il primo titolo riconosciuto vengono invece raccolti in una sezione priva di un'intestazione esplicita, identificata mediante un valore convenzionale. Titoli consecutivi con lo stesso testo normalizzato non determinano la creazione di sezioni distinte.

Per ciascuna sezione viene creato un *parent chunk*, contenente l'intero testo associato alla sezione. Quando è disponibile, il titolo viene anteposto al contenuto, così da rendere esplicito l'argomento anche nelle successive fasi di elaborazione. Il parent rappresenta pertanto l'unità documentale più ampia e può comprendere descrizioni, condizioni preliminari, passaggi operativi, note e avvertenze presenti tra due titoli consecutivi.

A ogni parent vengono associati un identificativo, il titolo della sezione, l'intervallo di pagine effettivamente coperto dal contenuto e i metadati relativi al capitolo e alla sezione ricostruiti durante il parsing. L'intervallo viene determinato considerando la pagina minima e quella massima dei blocchi appartenenti alla sezione.

Il contenuto del parent viene successivamente suddiviso in unità child più granulari. I blocchi vengono aggiunti progressivamente al child corrente finché la loro concatenazione non supera la dimensione massima configurata. Quando l'inserimento del blocco successivo eccederebbe tale limite, il child corrente viene chiuso e ne viene avviato uno nuovo.

Nella configurazione adottata, la dimensione massima predefinita di ciascun child è pari a 1800 caratteri, mentre la sovrapposizione tra frammenti consecutivi è pari a 150 caratteri. L'overlap viene ottenuto riportando all'inizio del nuovo child gli ultimi caratteri di quello precedente. Tale meccanismo riduce la probabilità che informazioni rilevanti poste in prossimità del confine vengano completamente separate, ma non rispetta necessariamente i limiti di parola o di frase.

Il limite dimensionale viene applicato durante l'aggregazione dei blocchi. Di conseguenza, un singolo blocco che superi da solo la dimensione massima non viene ulteriormente segmentato dalla funzione di chunking, ma viene mantenuto integralmente. Questo comportamento preserva l'unità del blocco estratto, pur potendo generare child più estesi rispetto alla soglia configurata.

Ogni child mantiene un riferimento esplicito al proprio parent mediante il campo `parent_id`. Sono inoltre conservati il titolo, l'intervallo di pagine e i metadati strutturali associati alla sezione. Le unità parent sono contrassegnate attraverso il campo booleano `is_parent`, mentre per i child tale valore è impostato a falso. Gli identificativi temporanei vengono successivamente tra-

sformati in UUID deterministici ottenuti a partire dal nome del documento e dall'identificativo locale del chunk.

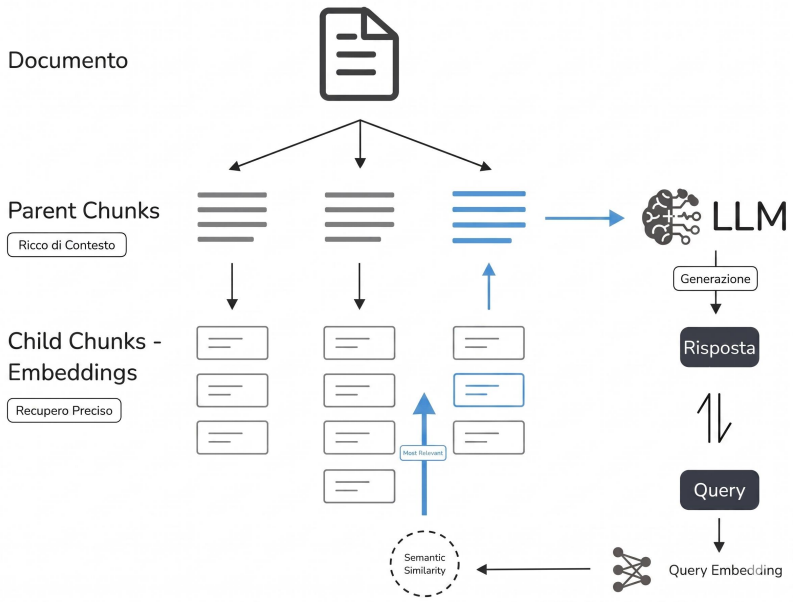


Figura 6.1: Rappresentazione della strategia parent/child

La pipeline evita inoltre la duplicazione delle sezioni brevi. Se la segmentazione produce un unico child il cui contenuto, comprensivo del titolo, coincide con quello del parent, tale child non viene aggiunto alla collezione. In questo caso il parent costituisce l'unica rappresentazione della sezione.

La rappresentazione gerarchica consente di separare l'unità utilizzata per localizzare l'informazione dall'unità destinata alla costruzione del contesto. I child offrono una rappresentazione più granulare delle sezioni estese e riducono il rischio che contenuti eterogenei vengano riassunti in un unico embedding. Il collegamento mediante `parent_id` consente successivamente di risalire dal frammento selezionato alla sezione più ampia dalla quale esso deriva.

Durante la fase online, i risultati granulari possono quindi essere materializzati recuperando i rispettivi parent. Più child riconducibili alla stessa sezione vengono associati alla medesima unità parent, che può essere inserita una sola volta nel contesto finale. In questo modo, il sistema può sfruttare unità specifiche per il retrieval e fornire al modello generativo sezioni maggiormente complete e contestualizzate.

6.3 Gestione dei metadati

A ciascun chunk prodotto dalla pipeline viene associato un insieme di metadati finalizzato a conservarne la provenienza, la collocazione all'interno del documento e le relazioni con le altre unità documentali. Tali informazioni sono mantenute separate dal contenuto testuale, ma vengono memorizzate insieme

a esso nella base di dati, così da poter essere utilizzate durante il retrieval, la materializzazione del contesto e la generazione della risposta.

La struttura dei metadati comprende un identificativo univoco, il titolo della sezione, il nome del file, la pagina iniziale e finale e, quando disponibili, il capitolo e la sezione di appartenenza. Sono inoltre presenti i campi `is_parent` e `parent_id`, utilizzati per rappresentare la relazione gerarchica tra le unità parent e child.

Per le unità parent, il campo `is_parent` è impostato a vero e `parent_id` non è valorizzato. I child sono invece contrassegnati mediante `is_parent = false` e conservano nel campo `parent_id` l'identificativo del parent di appartenenza. Questo collegamento consente, durante la fase online, di risalire dal frammento recuperato alla sezione documentale più ampia dalla quale esso deriva.

I campi `page_start` e `page_end` identificano l'intervallo di pagine coperto dal chunk. Tali informazioni permettono di ricondurre il contenuto alla posizione originale nel manuale e di includere nella risposta riferimenti verificabili. Per i parent, l'intervallo corrisponde alla pagina minima e massima dei blocchi appartenenti alla sezione; per i child, viene calcolato sulla base dei blocchi inclusi nel singolo frammento.

Le informazioni relative al titolo, al capitolo e alla sezione preservano invece il contesto logico del contenuto. Il titolo deriva dall'elemento utilizzato per delimitare la sezione, mentre capitolo e sezione vengono ricavati, quando possibile, dalle intestazioni presenti nella parte superiore delle pagine. Questi campi possono essere impiegati per presentare i risultati, applicare filtri e mantenere il collegamento tra il testo recuperato e l'organizzazione originaria del manuale.

La gestione dei metadati svolge pertanto quattro funzioni principali: garantire la tracciabilità dei contenuti, rappresentare la gerarchia parent/child, supportare la ricostruzione del contesto documentale e fornire i riferimenti necessari per la verifica delle risposte generate.

6.4 Indicizzazione in Qdrant

La fase di indicizzazione ha lo scopo di memorizzare le unità documentali prodotte durante il preprocessing in una struttura che consenta di eseguire sia ricerche semantiche sia ricerche lessicali. A tale scopo è stato utilizzato Qdrant, un database vettoriale che permette di associare a ciascun punto più rappresentazioni numeriche e un insieme di metadati descrittivi.

La collection viene configurata con due spazi di ricerca distinti, denominati rispettivamente `dense` e `bm25`. Lo spazio `dense` contiene gli embedding generati dal modello utilizzato dalla pipeline e adotta la distanza coseno come misura di confronto. La dimensionalità dei vettori viene ricavata dal modello di embedding e fornita al momento della creazione della collection, garantendo la coerenza tra le rappresentazioni prodotte e la configurazione dell'indice.

Lo spazio `bm25` è invece destinato alla ricerca lessicale e viene configurato come indice sparso con modificatore IDF. Per ciascun chunk, il testo viene fornito a Qdrant mediante un oggetto `Document` associato al modello `qdrant/bm25`. Prima dell'indicizzazione viene inoltre calcolata la lunghezza media dei conte-

nuti, espressa come numero di parole per chunk. Tale valore viene passato tra le opzioni del modello insieme all'indicazione della lingua inglese, così da supportare la normalizzazione del punteggio rispetto alla diversa estensione delle unità documentali.

Ogni unità parent e child viene memorizzata come un punto della collection. Il punto contiene il vettore denso, la rappresentazione lessicale BM25 e un payload composto dal contenuto testuale originale e dai relativi metadati. Nel payload vengono conservati l'identificativo, il titolo, il nome del documento, il capitolo, la sezione, l'intervallo di pagine, il campo `is_parent` e l'eventuale `parent_id`. Tali informazioni permettono di mantenere la tracciabilità dei contenuti e di ricostruire, durante la fase online, le relazioni gerarchiche tra i frammenti e le rispettive sezioni.

La presenza congiunta delle due rappresentazioni consente di utilizzare la medesima collection per modalità di ricerca differenti. La componente densa permette di individuare contenuti semanticamente affini alla richiesta, anche quando la formulazione utilizzata dall'utente differisce da quella presente nei manuali. La componente BM25 valorizza invece la corrispondenza lessicale e risulta particolarmente utile in presenza di termini tecnici, sigle, codici identificativi e denominazioni specifiche dei componenti. Le due modalità possono essere interrogate separatamente oppure combinate durante il retrieval ibrido.

L'inserimento dei punti viene effettuato in batch, così da ridurre il numero di operazioni inviate al database. Nella configurazione predefinita, i chunk vengono accumulati in gruppi di 128 elementi e caricati mediante operazioni di `upsert`. Al termine dell'elaborazione, anche l'eventuale gruppo residuo, composto da un numero inferiore di elementi, viene inserito nella collection.

Prima della creazione della collection, il sistema tenta di eliminare quella eventualmente già esistente con lo stesso nome. La collection viene quindi ricreata con la configurazione richiesta per gli indici denso e sparso. Durante gli esperimenti, tale comportamento garantisce che i punti presenti nel database derivino esclusivamente dalla configurazione di preprocessing e chunking sottoposta a valutazione, evitando commistioni con dati prodotti da esecuzioni precedenti.

Al termine dell'indicizzazione, Qdrant contiene pertanto una rappresentazione interrogabile della base documentale, composta dal testo dei chunk, dalle relative rappresentazioni dense e lessicali e dai metadati necessari a ricostruirne la provenienza e le relazioni gerarchiche. Questa struttura costituisce il punto di partenza per le modalità di retrieval dense, BM25 e ibride descritte nelle sezioni successive.

6.5 Retrieval ibrido

La fase di retrieval ha il compito di individuare, all'interno della collection Qdrant, i chunk maggiormente pertinenti rispetto alla domanda formulata dall'utente. La pipeline supporta tre modalità di ricerca: densa, lessicale mediante BM25 e ibrida. La configurazione principale utilizza il retrieval ibrido, con l'o-

biettivo di sfruttare le caratteristiche complementari della similarità semantica e della corrispondenza terminologica.

Prima dell'interrogazione della collection, la domanda originale può essere sottoposta a una fase di *query rewriting*. Tale operazione viene eseguita mediante un modello linguistico locale servito tramite Ollama, configurato per produrre fino a tre formulazioni alternative della richiesta. Le varianti devono risultare complementari: una deve rimanere vicina alla domanda originale, una deve adottare una formulazione più tecnica e compatta e una deve essere maggiormente orientata alle parole chiave.

Il prompt utilizzato per il query rewriting richiede di preservare esattamente l'intento informativo dell'utente, senza rispondere alla domanda né introdurre fatti, componenti, modelli o passaggi non implicati nella richiesta. Numeri, acronimi, nomi di componenti, codici di errore e unità di misura devono essere mantenuti invariati. L'output viene richiesto in formato JSON e comprende una descrizione sintetica dell'intento e la lista delle query generate.

Nella configurazione adottata, il query rewriter utilizza il modello `qwen3.5:4b`. La scelta di tale modello è motivata dal compromesso tra capacità linguistica e requisiti computazionali del sistema: il compito assegnato al rewriter non richiede generazione argomentativa complessa, ma la produzione controllata di poche riformulazioni semanticamente equivalenti o complementari alla query iniziale. Per questo motivo è stato privilegiato un modello di dimensioni contenute, eseguibile localmente e con costi di inferenza ridotti, rispetto a modelli di taglia superiore.

La temperatura è stata impostata a 0.1 al fine di ridurre la variabilità delle riformulazioni generate e aumentare la stabilità del comportamento del sistema. Inoltre, il ragionamento esplicito è stato disabilitato, poiché l'obiettivo del componente non è produrre spiegazioni o catene inferenziali, ma generare query brevi e direttamente utilizzabili dal modulo di retrieval. Dopo la generazione, le query vuote, duplicate o coincidenti con la domanda originale vengono eliminate. Sono mantenute al massimo tre riformulazioni, successivamente fornite al componente di retrieval insieme alla richiesta iniziale.

Nella ricerca densa, ciascuna query viene trasformata in un embedding mediante lo stesso modello E5 utilizzato durante l'indicizzazione. Prima della codifica viene aggiunto il prefisso `query:`, coerentemente con il formato previsto dal modello, e il vettore risultante viene normalizzato. La ricerca viene quindi eseguita nello spazio `dense` della collection, configurato con distanza coseno.

La ricerca densa consente di recuperare contenuti semanticamente affini anche quando la formulazione della domanda differisce da quella presente nel manuale. Una richiesta espressa in linguaggio naturale può quindi essere associata a una procedura descritta mediante termini più tecnici o attraverso una formulazione differente.

La ricerca lessicale utilizza invece il modello `qdrant/bm25`. La query viene fornita come documento testuale, specificando l'inglese come lingua di riferimento, e confrontata con le rappresentazioni sparse memorizzate nello spazio `bm25`. Tale modalità risulta particolarmente utile nel dominio considerato, nel quale le richieste possono contenere sigle, codici identificativi, valori numerici, deno-

minazioni di componenti o termini tecnici che richiedono una corrispondenza lessicale precisa.

Nella modalità ibrida, per ciascuna formulazione vengono eseguite entrambe le modalità di ricerca. Con la sola query originale vengono quindi prodotte una graduatoria densa e una graduatoria BM25. In presenza di query alternative, ciascuna formulazione genera a sua volta una ricerca densa e una ricerca lessicale. Tutte le graduatorie ottenute vengono successivamente aggregate mediante Reciprocal Rank Fusion (RRF).

La RRF assegna rilevanza ai risultati sulla base delle posizioni occupate nelle diverse graduatorie, senza richiedere la normalizzazione diretta dei punteggi prodotti dai retriever. In questo modo, un chunk può essere valorizzato sia perché compare nelle prime posizioni di una singola ricerca, sia perché viene recuperato in modo consistente da più query o modalità di retrieval.

L'utilizzo congiunto della domanda originale e delle riformulazioni mira ad aumentare la probabilità di individuare la procedura corretta quando il lessico impiegato dall'utente non coincide con quello adottato dal costruttore. Il mantenimento della query iniziale riduce inoltre il rischio che un'eventuale riformulazione alteri o restringa impropriamente l'intento della richiesta.

La Figura 6.2 schematizza il flusso del retrieval ibrido, dalla generazione delle query alternative alla fusione delle graduatorie dense e BM25.

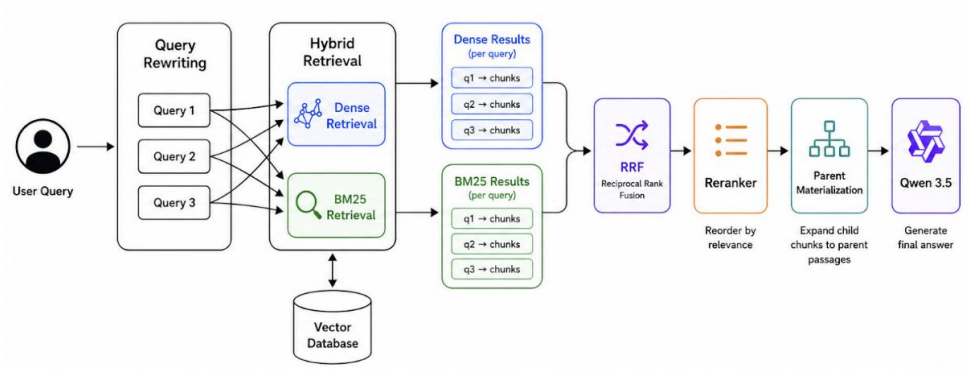


Figura 6.2: Flusso del retrieval ibrido

La ricerca recupera inizialmente un insieme di candidati più ampio rispetto al numero di risultati finali richiesti. Il numero di candidati restituiti dalla fusione è impostato al valore maggiore tra il limite finale richiesto e 20. Per ciascuna ricerca preliminare, il numero massimo di elementi recuperabili è invece pari al valore maggiore tra 100 e cinque volte il numero dei candidati.

Indicando con (k) il numero di risultati finali richiesti, i parametri sono definiti come:

$$k_{\text{cand}} = \max(k, 20)$$

$$k_{\text{prefetch}} = \max(100, 5k_{\text{cand}})$$

Tale configurazione fornisce alle fasi successive un insieme sufficientemente ampio di contenuti da confrontare, mantenendo distinto il numero dei candidati iniziali dal numero di risultati effettivamente selezionati.

Il retrieval ibrido combina pertanto la capacità della ricerca densa di riconoscere affinità semantiche con la precisione terminologica di BM25. Il query rewriting amplia ulteriormente le modalità con cui l'esigenza informativa può essere rappresentata, mentre la fusione mediante RRF produce una graduatoria unificata di chunk candidati, successivamente sottoposta alle fasi di reranking e materializzazione.

6.6 Reranking e materializzazione

Al termine della ricerca, il sistema dispone di un insieme di chunk candidati ordinati secondo i punteggi prodotti dal retrieval. Nella configurazione ibrida, tale graduatoria deriva dalla fusione mediante Reciprocal Rank Fusion dei risultati ottenuti dalle componenti densa e BM25. L'ordinamento risultante consente di selezionare contenuti potenzialmente pertinenti, ma non rappresenta necessariamente la valutazione più accurata della relazione tra la domanda e ciascun frammento. Per migliorare la precisione dei risultati nelle prime posizioni viene pertanto applicata una fase di reranking.

Il reranking utilizza un modello Cross-Encoder, che riceve congiuntamente la domanda e il contenuto di ciascun chunk candidato. A differenza del retrieval denso, nel quale query e documenti vengono codificati separatamente e confrontati nello spazio vettoriale, il Cross-Encoder elabora direttamente la coppia formata dai due testi e produce un punteggio di rilevanza specifico. Tale valutazione è generalmente più onerosa dal punto di vista computazionale e viene quindi applicata soltanto all'insieme ristretto di candidati precedentemente recuperati.

Per ciascun candidato viene costruita una coppia composta dalla domanda originale dell'utente e dal contenuto memorizzato nel payload del chunk. Il reranker assegna un punteggio a ogni coppia e i risultati vengono riordinati in senso decrescente. Anche quando il retrieval è stato eseguito utilizzando formulazioni alternative della query, la fase di reranking considera la richiesta originale, così da valutare la pertinenza rispetto all'intento espresso direttamente dall'utente.

Dopo il riordinamento vengono mantenuti soltanto i primi risultati previsti dalla configurazione. In assenza del Cross-Encoder, la pipeline conserva invece l'ordinamento prodotto dal retrieval iniziale e applica direttamente lo stesso limite al numero di elementi selezionati.

I chunk maggiormente rilevanti possono tuttavia rappresentare soltanto porzioni limitate delle rispettive sezioni. Sebbene tale granularità favorisca la precisione della ricerca, un frammento potrebbe non contenere tutte le condizioni preliminari, le avvertenze o le operazioni necessarie per interpretare correttamente una procedura. Per questo motivo, dopo la selezione viene applicata, quando abilitata, la materializzazione delle unità parent.

Quando un risultato corrisponde a un child e contiene un valore valido nel campo `parent_id`, il sistema recupera da Qdrant il relativo parent. Il contenuto del child viene sostituito, nell'output del retrieval, con il testo completo della sezione parent e con i relativi metadati. Il punteggio associato al risultato rimane quello attribuito al child dal reranker oppure, in assenza di reranking, quello restituito dalla ricerca iniziale. In questo modo, la rilevanza viene determinata sulla base del frammento specifico, mentre il contesto fornito alle fasi successive corrisponde a un'unità documentale più ampia.

Gli identificativi dei parent associati ai risultati selezionati vengono raccolti in un insieme e recuperati mediante un'unica operazione verso Qdrant. I payload ottenuti vengono organizzati in una struttura indicizzata per identificativo, così da consentire l'associazione tra ciascun child e il relativo parent senza effettuare una richiesta distinta per ogni risultato.

Più child presenti tra i risultati possono appartenere alla stessa sezione. La loro materializzazione produrrebbe quindi copie ripetute del medesimo parent. Per evitare tale ridondanza, la pipeline mantiene l'insieme degli identificativi già inseriti e include ciascun parent una sola volta nell'output finale. I child successivi associati alla stessa unità vengono ignorati.

Se un risultato è già un parent, viene mantenuto direttamente, evitando eventuali duplicazioni. Analogamente, se un chunk non possiede un riferimento gerarchico valido o il relativo parent non viene recuperato, il contenuto originale del risultato viene conservato senza sostituzioni. La pipeline permette inoltre di disabilitare la materializzazione mediante un parametro di configurazione, restituendo in tal caso direttamente i chunk selezionati e i relativi metadati.

La selezione dei primi (k) candidati avviene prima della deduplicazione dei parent. Di conseguenza, se più risultati compresi nelle prime (k) posizioni fanno riferimento alla stessa sezione, il numero di unità finali materializzate può essere inferiore al limite richiesto, poiché la pipeline non recupera ulteriori candidati per sostituire i duplicati eliminati.

La combinazione tra reranking e materializzazione separa pertanto due funzioni complementari. Il reranker migliora l'ordinamento dei frammenti candidati sulla base della loro pertinenza rispetto alla domanda, mentre la materializzazione amplia i risultati selezionati recuperando il relativo contesto documentale. Il sistema può così effettuare la ricerca su unità granulari e fornire al modello generativo sezioni più complete, mantenendo al contempo la relazione con il frammento che ne ha determinato la selezione.

6.7 Prompt engineering

La fase di prompt engineering ha lo scopo di guidare il modello linguistico nella produzione di risposte tecniche, concise e aderenti alla documentazione recuperata. Il prompt viene costruito dinamicamente a partire dalla domanda originale dell'utente e dai risultati selezionati durante le precedenti fasi di retrieval, reranking e materializzazione.

Per ciascun risultato viene costruito un blocco di contesto numerato, contenente il titolo della sezione, la pagina iniziale associata al contenuto e il relativo

testo. I diversi blocchi vengono separati mediante righe vuote e identificati progressivamente come *Source 1*, *Source 2* e così via. Tale organizzazione consente al modello di distinguere le informazioni provenienti da fonti differenti e di utilizzare i relativi metadati nella costruzione delle citazioni.

Il prompt assegna al modello il ruolo di assistente tecnico specializzato in manuali e procedure di manutenzione. Le istruzioni richiedono di rispondere utilizzando soltanto il contesto recuperato, evitando il ricorso a conoscenze esterne e l'introduzione di passaggi, avvertenze, cause, valori o ipotesi non supportati dai documenti forniti.

Qualora il contesto non contenga informazioni sufficienti per formulare una risposta affidabile, il modello viene istruito a restituire esattamente la seguente frase:

```
The answer is not available in the provided context.
```

L'impiego di una risposta predefinita consente di rendere più riconoscibili e uniformi i casi nei quali le informazioni recuperate risultano insufficienti, evitando che il modello tenti di completare la risposta mediante supposizioni.

La struttura dell'output dipende dalla natura della richiesta. Se la domanda riguarda una procedura, il modello deve presentare le operazioni mediante una lista numerata. Per le richieste descrittive deve invece produrre un breve paragrafo. In entrambi i casi, la risposta deve mantenere uno stile conciso, tecnico e preciso.

Ogni affermazione ricavata dal contesto deve essere accompagnata da una citazione nel formato `[title, page]`, collocata al termine della frase pertinente. Il modello può utilizzare esclusivamente i titoli e i numeri di pagina presenti nei blocchi di contesto. Nell'implementazione corrente, il valore riportato nel prompt corrisponde al campo `page_start` del risultato recuperato; l'eventuale pagina finale non viene quindi inclusa direttamente nelle istruzioni fornite al modello.

Il prompt richiede infine di restituire esclusivamente la risposta finale, senza preamboli generici, spiegazioni sul funzionamento del sistema o descrizioni del processo di ragionamento. La generazione viene eseguita tramite Ollama, utilizzando come configurazione predefinita il modello `qwen3.5:4b`. La temperatura è impostata a (0.2), così da contenere la variabilità dell'output, mentre l'opzione `think` è disabilitata.

La richiesta viene inviata all'endpoint locale di Ollama in modalità non streaming. La risposta restituita dal servizio viene letta dal campo `response`, ripulita dagli spazi iniziali e finali e utilizzata come output conclusivo della pipeline.

L'insieme di questi vincoli non elimina completamente la possibilità di errori, ma limita la libertà generativa del modello e orienta la risposta verso le informazioni effettivamente recuperate. L'efficacia del prompt rimane tuttavia dipendente dalla pertinenza, dalla completezza e dalla corretta materializzazione del contesto prodotto nelle fasi precedenti.

6.8 Esposizione pipeline tramite servizi API

Oltre alla progettazione delle singole componenti di elaborazione, la pipeline è stata organizzata in modo da poter essere invocata mediante servizi API. Tale scelta consente di separare la logica interna del sistema dalle modalità con cui esso viene utilizzato da eventuali applicazioni esterne, interfacce utente o strumenti di valutazione sperimentale.

L'esposizione tramite API riguarda principalmente due fasi del processo. La prima è la fase di ingestion, eseguita offline, nella quale un manuale tecnico viene acquisito, elaborato, suddiviso in unità documentali e indicizzato. La seconda è la fase di interrogazione, eseguita online, nella quale una domanda formulata dall'utente viene utilizzata per recuperare i contenuti pertinenti e generare una risposta fondata sulla documentazione disponibile.

Questa organizzazione consente di mantenere una distinzione esplicita tra la preparazione della base documentale e l'utilizzo della pipeline in fase di consultazione. La fase di ingestion può essere eseguita ogni volta che viene aggiunto o aggiornato un manuale, mentre la fase di risposta può essere richiamata ripetutamente dagli utenti finali o dagli script di valutazione.

6.8.1 Servizio di ingestion

Il servizio di ingestion ha il compito di avviare il processo di elaborazione dei manuali tecnici. L'input principale è costituito dal file PDF del manuale, eventualmente accompagnato da metadati descrittivi, quali il nome del documento, la categoria dell'apparato o l'identificativo della configurazione sperimentale.

Una volta ricevuto il documento, il servizio attiva le componenti descritte nelle sezioni precedenti: parsing del PDF, filtraggio degli elementi non rilevanti, ricostruzione della struttura documentale, generazione delle unità parent e child, calcolo delle rappresentazioni dense e sparse e caricamento dei punti nella collection Qdrant. Al termine dell'operazione, il servizio restituisce informazioni relative all'esito dell'elaborazione, come il numero di chunk prodotti, la collection aggiornata e l'eventuale presenza di errori.

Nel caso di elaborazioni più onerose, l'ingestion può essere gestita come un job asincrono. In questa configurazione, una prima chiamata avvia il processo e restituisce un identificativo del job, mentre chiamate successive permettono di verificarne lo stato. Tale modalità risulta utile quando i manuali hanno dimensioni elevate o quando l'elaborazione richiede operazioni costose, come il calcolo degli embedding e l'indicizzazione.

Endpoint	Metodo	Funzione	Input
`/documents`	POST	Registra e indicizza un nuovo manuale	`title`, `url`
`/documents`	GET	Restituisce l'elenco dei manuali registrati	`status` opzionale
`/documents/{document_id}`	GET	Recupera i metadati di un manuale	`document_id`
`/documents/{document_id}`	DELETE	Elimina un manuale indicizzato	`document_id`

Tabella 6.1: Endpoint del servizio `documents`.

6.8.2 Servizio di generazione della risposta

Il servizio di interrogazione rappresenta il punto di accesso alla pipeline durante la fase online. L'utente, o un'applicazione esterna, invia una domanda in linguaggio naturale e specifica, quando necessario, il documento, la collection o l'insieme di manuali sui quali eseguire la ricerca.

La richiesta viene elaborata seguendo il flusso descritto nelle sezioni precedenti. La domanda può essere sottoposta a query rewriting, quindi viene utilizzata per interrogare la collection mediante retrieval ibrido. I risultati ottenuti dalla ricerca densa e da quella lessicale vengono combinati, riordinati tramite reranking e, quando corrispondono a unità child, materializzati recuperando le rispettive unità parent. Il contesto finale viene infine inserito nel prompt fornito al modello linguistico, che genera la risposta tecnica.

L'output del servizio non comprende soltanto il testo generato, ma anche i riferimenti alle fonti utilizzate. In particolare, la risposta può essere accompagnata dai titoli delle sezioni, dai nomi dei documenti, dagli intervalli di pagina e dagli identificativi dei chunk o dei parent recuperati. Queste informazioni consentono di verificare il contenuto prodotto mediante la consultazione diretta del manuale originale.

Endpoint	**Metodo**	**Funzione**	**Input**
`/chat`	POST	Interroga un manuale indicizzato e genera una risposta RAG	`title`, `query`

Tabella 6.2: Endpoint del servizio `chat`.

Capitolo 7

Valutazione sperimentale

Il presente capitolo descrive la metodologia adottata per valutare le diverse configurazioni analizzate nel corso del lavoro. L'obiettivo della sperimentazione è misurare separatamente la qualità del retrieval e quella della risposta generata, così da distinguere gli errori dovuti al mancato recupero delle informazioni da quelli introdotti durante la fase di generazione.

La valutazione è stata condotta su un dataset costruito manualmente a partire da manuali tecnici reali. Per ciascuna domanda sono stati annotati la procedura di riferimento, il relativo contenuto testuale e le pagine del documento nelle quali essa compare. Tale struttura consente di confrontare in modo controllato le differenti strategie di preprocessing, chunking e retrieval.

La qualità del retrieval viene analizzata secondo due prospettive complementari. La prima riguarda la capacità del sistema di recuperare le pagine corrette; la seconda considera la corrispondenza tra il contenuto selezionato e la procedura di riferimento. Questa distinzione permette di valutare separatamente la localizzazione documentale e l'effettiva rilevanza del testo recuperato.

La fase di generazione viene invece valutata considerando la qualità della risposta prodotta a partire dal contesto selezionato. L'analisi prende in esame la correttezza, la completezza e la fedeltà rispetto alla documentazione di riferimento, con particolare attenzione alla presenza di informazioni non supportate o all'omissione di passaggi rilevanti.

Le sezioni successive descrivono il dataset utilizzato, le metriche adottate per la valutazione del retrieval e della generazione e le baseline considerate nel confronto sperimentale.

7.1 Dataset di valutazione

Per confrontare in modo controllato le diverse strategie è stato costruito manualmente un dataset di riferimento a partire dagli otto manuali tecnici considerati nel progetto. Il dataset comprende complessivamente 70 esempi, distribuiti tra documenti relativi a differenti categorie di apparati, tra cui motori, pompe, sistemi per la produzione di acqua calda, impianti di ventilazione, sistemi idrici pressurizzati e stabilizzatori.

Ciascun esempio associa una domanda formulata in linguaggio naturale a una specifica procedura presente nel relativo manuale. La procedura di riferimento è stata individuata mediante consultazione diretta del documento e trascritta integralmente, includendo, quando presenti, passaggi numerati, condizioni preliminari, note, avvertenze e riferimenti interni. Sono state inoltre annotate tutte le pagine necessarie a ricostruire il contenuto completo della procedura.

Il dataset è organizzato in file JSON Lines, ciascuno dei quali raccoglie gli esempi relativi a uno specifico manuale. Durante la valutazione, i file con estensione `.jsonl` vengono individuati automaticamente all'interno della directory dedicata e letti riga per riga. Le pagine annotate vengono normalizzate come valori interi prima del calcolo delle metriche.

Ogni esempio comprende i seguenti campi:

- `id`: identificativo dell'esempio;
- `manual`: nome del manuale PDF di riferimento;
- `procedure_title`: titolo della procedura attesa;
- `procedure_text`: contenuto completo della procedura di riferimento;
- `pages`: elenco ordinato delle pagine nelle quali la procedura compare;
- `query`: domanda in linguaggio naturale associata alla procedura.

Le query sono state formulate in modo sintetico e non coincidono necessariamente con il titolo della procedura o con le espressioni utilizzate dal costruttore. Tale differenza consente di verificare se il sistema sia in grado di collegare una richiesta espressa in linguaggio naturale alla terminologia specialistica adottata nella documentazione.

Le procedure annotate possono essere contenute in una singola pagina oppure estendersi su più pagine. Alcune operazioni brevi, come la calibrazione di un indicatore, richiedono una sola pagina; procedure più articolate, come l'avviamento in particolari condizioni operative o la sostituzione di un componente, possono invece coinvolgere più pagine consecutive. L'annotazione non riguarda pertanto soltanto la pagina nella quale compare il titolo, ma tutte le pagine necessarie a ricostruire la procedura di riferimento.

Il titolo, il testo e le pagine annotate costituiscono la ground truth utilizzata nella valutazione. Le pagine permettono di misurare la capacità del sistema di localizzare correttamente la procedura all'interno del manuale. Il contenuto testuale consente invece di confrontare i risultati recuperati con la procedura attesa, valutandone la precisione lessicale, la copertura e la similarità complessiva.

Durante l'esecuzione sperimentale, il nome del manuale associato a ciascun esempio viene confrontato con il metadato `filename` dei risultati recuperati. Il confronto viene effettuato dopo aver rimosso l'estensione e normalizzato il nome in minuscolo, così da evitare che risultati provenienti da documenti differenti vengano considerati corretti soltanto perché associati alle stesse pagine.

La costruzione manuale del dataset consente quindi di valutare non soltanto se il sistema recuperi una sezione genericamente pertinente, ma se riesca a individuare la procedura attesa nel manuale corretto e con un contesto sufficientemente completo per rispondere alla domanda.

7.2 Metriche di retrieval

La valutazione del retrieval è stata articolata su due livelli complementari. Il primo riguarda la capacità del sistema di individuare le pagine nelle quali è contenuta la procedura corretta; il secondo misura la corrispondenza tra il contenuto recuperato e la procedura annotata nel dataset.

Per ciascuna domanda vengono considerati i primi tre risultati restituiti dal sistema. Tale scelta consente di valutare non soltanto il contenuto collocato in prima posizione, ma anche la capacità della pipeline di includere la procedura corretta tra i risultati maggiormente rilevanti.

Le metriche sulle pagine verificano se almeno uno dei risultati recuperati copra le pagine annotate nella ground truth e in quale posizione compaia il primo risultato rilevante. Le metriche sul contenuto confrontano invece il testo dei risultati con la procedura di riferimento, considerando sia la sovrapposizione lessicale sia differenti misure di similarità testuale e semantica.

Le due prospettive permettono di distinguere casi differenti. Un risultato può infatti provenire dalle pagine corrette ma contenere soltanto una parte della procedura, oppure può presentare un testo semanticamente affine senza corrispondere esattamente alla sezione annotata. La valutazione congiunta consente quindi di analizzare sia la localizzazione documentale sia la qualità effettiva del contenuto recuperato.

7.2.1 Metriche sulle pagine

Le pagine associate a ciascun risultato vengono ricavate dai metadati `page_start` e `page_end`. Quando un contenuto si estende su più pagine, l'insieme delle pagine previste viene ottenuto considerando tutti i valori interi compresi nell'intervallo. Qualora la pagina finale risulti inferiore a quella iniziale, i due estremi vengono scambiati prima della costruzione dell'insieme.

Il confronto viene effettuato soltanto per i risultati appartenenti al manuale corretto. A tale scopo, il nome del documento memorizzato nei metadati viene normalizzato, rimuovendo l'estensione e convertendolo in minuscolo, e confrontato con il campo `manual` della ground truth. In questo modo, la coincidenza dei numeri di pagina tra manuali differenti non determina erroneamente un risultato positivo.

La **Page Precision** misura la proporzione delle pagine associate a un risultato che appartiene anche all'insieme delle pagine attese. Indicando con (P_r) l'insieme delle pagine del risultato recuperato e con (P_g) l'insieme delle pagine annotate nella ground truth, la metrica è definita come:

$$\text{PagePrecision}(P_r, P_g) = \frac{|P_r \cap P_g|}{|P_r|}$$

Un valore elevato indica che l'intervallo recuperato contiene poche pagine estranee rispetto alla procedura cercata. Se il risultato non contiene metadati di pagina validi, il relativo punteggio non viene considerato tra i candidati utili e, in assenza di altri risultati validi, la precisione assume valore nullo.

La **Page Recall** misura invece la quota delle pagine attese inclusa nel risultato:

$$\text{PageRecall}(P_r, P_g) = \frac{|P_r \cap P_g|}{|P_g|}$$

Questa metrica risulta particolarmente importante per le procedure distribuite su più pagine, poiché permette di verificare se il risultato copra l'intero intervallo necessario oppure soltanto una sua parte.

Per ciascuna domanda, Page Precision e Page Recall vengono calcolate separatamente sui primi tre risultati. Il punteggio associato alla query corrisponde al valore massimo ottenuto da un singolo risultato appartenente al manuale corretto:

$$\text{PagePrecision@3} = \max_{r \in R_3} \text{PagePrecision}(P_r, P_g)$$

$$\text{PageRecall@3} = \max_{r \in R_3} \text{PageRecall}(P_r, P_g)$$

dove R_3 rappresenta l'insieme dei primi tre risultati restituiti. Le due metriche non vengono quindi calcolate sull'unione delle pagine presenti nella top-3, ma verificano se almeno uno dei risultati recuperati presenta un intervallo preciso o sufficientemente completo.

PagePrecision@3 e PageRecall@3 possono inoltre selezionare risultati differenti. Un frammento breve può ottenere la precisione migliore perché contiene soltanto pagine corrette, mentre un risultato più ampio può ottenere il richiamo migliore perché copre una quota maggiore delle pagine annotate.

Il **Mean Reciprocal Rank** (MRR) valuta invece la posizione del primo risultato rilevante. Un risultato è considerato rilevante quando appartiene al manuale corretto e il relativo intervallo condivide almeno una pagina con la ground truth. Per una singola domanda, il Reciprocal Rank è definito come:

$$\text{RR} = \begin{cases} \dfrac{1}{r}, & \text{se esiste un risultato rilevante;} \\ 0, & \text{altrimenti;} \end{cases}$$

dove r indica la posizione del primo risultato rilevante. Se il primo risultato condivide almeno una pagina corretta, il punteggio è pari a 1; se il primo risultato rilevante compare in seconda o terza posizione, il punteggio è rispettivamente pari a $\frac{1}{2}$ o $\frac{1}{3}$.

L'MRR complessivo viene calcolato come media dei Reciprocal Rank sulle (N) domande del dataset:

$$\text{MRR} = \frac{1}{N} \sum_{i=1}^{N} \text{RR}_i$$

A differenza di Page Precision e Page Recall, l'MRR non valuta la completezza dell'intervallo recuperato: è sufficiente che il primo risultato rilevante condivida almeno una pagina con quelle annotate.

7.2.2 Metriche sul contenuto recuperato

Il recupero delle pagine corrette non garantisce che il testo selezionato coincida con la procedura attesa. Un risultato può infatti provenire dall'intervallo corretto ma contenere informazioni estranee, oppure può riportare soltanto una parte della procedura. Per questo motivo, il contenuto dei primi tre risultati viene confrontato con il campo `procedure_text` del dataset mediante metriche lessicali, sequenziali e semantiche.

Prima del calcolo delle metriche lessicali, i testi vengono normalizzati. Tutti i caratteri sono convertiti in minuscolo, le sequenze di spazi vengono uniformate e i simboli diversi da lettere, cifre, lettere accentate, apostrofi e spazi vengono rimossi. Il testo risultante viene quindi suddiviso in token sulla base degli spazi.

La **Token Precision** misura la proporzione delle occorrenze lessicali presenti nel risultato che trovano corrispondenza nella procedura di riferimento. Indicando con $C_r(t)$ e $C_g(t)$ il numero di occorrenze del token (t) rispettivamente nel risultato e nella ground truth, il numero complessivo di occorrenze condivise è:

$$O = \sum_{t \in V} \min\left(C_r(t), C_g(t)\right)$$

La Token Precision è quindi definita come:

$$\text{TokenPrecision} = \frac{O}{|T_r|}$$

dove $|T_r|$ è il numero complessivo di token del risultato. Un valore elevato indica che il contenuto recuperato presenta una quantità limitata di testo non condiviso con la procedura attesa.

La **Token Recall** misura invece la proporzione delle occorrenze presenti nella procedura di riferimento che sono state recuperate:

$$\text{TokenRecall} = \frac{O}{|T_g|}$$

dove $|T_g|$ rappresenta il numero totale di token della ground truth. Un valore elevato indica che il risultato copre una parte consistente del contenuto atteso.

La **Token F1** è la media armonica di precision e recall:

$$\text{TokenF1} = 2\frac{\text{TokenPrecision} \cdot \text{TokenRecall}}{\text{TokenPrecision} + \text{TokenRecall}}$$

Quando entrambe le metriche hanno valore nullo, anche il Token F1 viene posto a zero. Per ciascuna domanda, tra i primi tre risultati viene selezionato quello che ottiene il valore più elevato di Token F1; precision e recall aggregate per la query sono quindi quelle appartenenti allo stesso risultato scelto sulla base dell'F1.

La **Sequence Similarity** confronta la sequenza normalizzata dei token del risultato con quella della procedura di riferimento. Dopo la normalizzazione, i token vengono ricomposti in una stringa e confrontati mediante l'algoritmo `SequenceMatcher`. La metrica tiene conto della presenza e dell'ordine delle sottosequenze comuni e risulta pertanto sensibile alla disposizione del contenuto, oltre che alla semplice sovrapposizione lessicale.

La **Embedding Similarity** misura invece la vicinanza semantica tra il risultato e la procedura attesa. I due testi vengono codificati mediante il modello di embedding utilizzato dalla pipeline, producendo vettori normalizzati. Il punteggio è calcolato tramite prodotto scalare:

$$\text{EmbeddingSimilarity}(r, g) = \mathbf{e}_r^\top \mathbf{e}_g$$

Poiché i vettori sono normalizzati, il prodotto scalare coincide con la similarità coseno. Questa metrica può riconoscere una vicinanza semantica anche quando risultato e riferimento impiegano formulazioni lessicalmente differenti.

Il **BERTScore F1** valuta infine la corrispondenza tra risultato e riferimento mediante rappresentazioni contestuali dei token. I token dei due testi vengono allineati in base alla similarità delle rispettive rappresentazioni, consentendo di riconoscere termini e formulazioni semanticamente affini anche in assenza di una corrispondenza esatta. Il valore F1 combina la precisione e il richiamo ottenuti da tale allineamento contestuale.

Per Sequence Similarity, Embedding Similarity e BERTScore F1 viene selezionato separatamente il valore massimo tra i primi tre risultati:

$$M@3 = \max_{r \in R_3} M(r, g)$$

dove M rappresenta una delle tre metriche e g la procedura di riferimento. Di conseguenza, il risultato migliore può essere differente per ciascuna misura: il chunk che massimizza la similarità sequenziale non coincide necessariamente con quello che massimizza la similarità degli embedding o il BERTScore.

I punteggi complessivi sono ottenuti calcolando la media dei valori per-query su tutte le domande del dataset. In assenza di risultati, le metriche lessicali, sequenziali e di embedding assumono valore zero.

Oltre alle metriche di qualità, viene registrata la latenza della fase di retrieval. La misurazione ha inizio immediatamente prima della chiamata al metodo di ricerca e termina al completamento della stessa. Il valore comprende quindi la ricerca in Qdrant, l'eventuale fusione delle graduatorie, il reranking e la materializzazione dei parent, ma non include la generazione della risposta da parte del modello linguistico. La latenza media è calcolata sull'insieme delle query e consente di analizzare il compromesso tra qualità del recupero e costo computazionale delle differenti configurazioni.

7.3 Metriche di generazione

La valutazione della fase di generazione ha lo scopo di verificare se il modello linguistico produca risposte corrette, complete e supportate dai contenuti recu-

perati. Tale analisi viene condotta separatamente dalla valutazione del retrieval, poiché la presenza della procedura corretta nel contesto non garantisce che il modello la utilizzi in modo appropriato durante la costruzione della risposta.

Per ciascuna domanda, il modello generativo riceve i risultati selezionati dalle fasi di retrieval, reranking e materializzazione. La risposta prodotta viene successivamente confrontata sia con la procedura di riferimento annotata nel dataset, sia con il contesto effettivamente fornito al modello. Questa doppia prospettiva consente di distinguere gli errori rispetto alla risposta attesa dalle informazioni eventualmente introdotte senza supporto documentale.

La valutazione qualitativa è basata sul paradigma *LLM-as-a-Judge*, nel quale un modello linguistico distinto da quello sottoposto a valutazione assume il ruolo di giudice automatico [23]. Il giudice riceve quattro elementi:

- la domanda formulata dall'utente;
- il contesto recuperato dalla pipeline;
- la procedura di riferimento annotata nel dataset;
- la risposta generata dal sistema.

Sulla base di tali informazioni, il giudice assegna un punteggio compreso tra 1 e 5 a tre dimensioni qualitative: *correctness*, *faithfulness* e *completeness*. Un valore pari a 1 indica prestazioni insufficienti rispetto al criterio considerato, mentre un valore pari a 5 identifica una risposta pienamente soddisfacente.

La **correctness** misura la correttezza complessiva della risposta rispetto alla domanda e alla procedura di riferimento. Una risposta riceve un punteggio elevato quando fornisce le informazioni richieste, conserva il significato tecnico della documentazione e non contiene affermazioni errate o contraddittorie. Questo criterio verifica quindi se il contenuto generato risponda effettivamente all'esigenza informativa dell'utente.

La **faithfulness** valuta la fedeltà della risposta al contesto recuperato. Il giudice verifica che le affermazioni prodotte siano supportate dai contenuti forniti al modello e che non siano stati aggiunti passaggi operativi, valori, cause, condizioni o avvertenze non presenti nelle fonti selezionate. Tale dimensione è finalizzata a rilevare contenuti non documentati, anche quando risultano plausibili dal punto di vista linguistico o tecnico.

La **completeness** misura il grado con cui la risposta include gli elementi rilevanti presenti nella procedura di riferimento. Nel caso di richieste procedurali, vengono considerati i passaggi operativi, le condizioni preliminari, le note, le avvertenze e gli eventuali controlli conclusivi. Una risposta può pertanto risultare corretta e fedele al contesto, ma ricevere un punteggio di completezza inferiore qualora ometta una parte significativa della procedura.

Le tre dimensioni risultano complementari. La correctness valuta l'adeguatezza della risposta rispetto alla domanda e al riferimento; la faithfulness controlla che ogni informazione sia supportata dal contesto recuperato; la completeness misura la copertura degli elementi attesi. La loro valutazione separata permette, ad esempio, di distinguere una risposta corretta ma incompleta da una risposta dettagliata che include informazioni non presenti nelle fonti.

Per ciascuna dimensione, il punteggio complessivo viene ottenuto calcolando la media aritmetica dei valori assegnati alle N risposte valutate:

$$\overline{s}_m = \frac{1}{N} \sum_{i=1}^{N} s_{i,m}$$

dove $s_{i,m}$ rappresenta il punteggio assegnato alla risposta i-esima per la metrica m, con $m \in$ correctness, faithfulness, completeness.

Nel presente lavoro, la valutazione della generazione è stata applicata alla configurazione completa della pipeline personalizzata, comprendente retrieval ibrido, reranking e materializzazione dei parent. La risposta è stata prodotta utilizzando il contesto restituito da tale configurazione e confrontata con la procedura di riferimento associata alla medesima domanda.

L'impiego di un LLM come giudice consente di valutare proprietà difficilmente rappresentabili mediante sole metriche basate sulla sovrapposizione lessicale. I punteggi ottenuti dipendono tuttavia dal modello valutatore, dal prompt di giudizio e dalla scala adottata. Essi devono pertanto essere interpretati come una misura automatica comparativa e non come una sostituzione completa della valutazione umana.

7.4 Baseline di confronto

La valutazione sperimentale ha coinvolto le strategie di preprocessing e chunking descritte nel Capitolo 5. La pipeline aziendale *h_service* è stata adottata come baseline principale, poiché rappresentava la soluzione disponibile prima dello sviluppo del presente lavoro.

I risultati della baseline sono stati confrontati con quelli ottenuti mediante l'approccio basato su Unstructured, la pipeline personalizzata e la strategia basata sulla conversione dei documenti tramite MinerU. È stato inoltre considerato un approccio black-box basato su un modello multimodale della famiglia Gemini, nel quale il manuale PDF viene fornito direttamente al modello senza una fase esplicitamente implementata di indicizzazione e retrieval.

Per garantire un confronto omogeneo tra le pipeline RAG, sono stati mantenuti invariati il dataset di valutazione, le metriche, il modello di embedding, il database vettoriale e la configurazione delle fasi successive al preprocessing. Le diverse strategie sono state quindi valutate utilizzando la medesima modalità di retrieval ibrido, lo stesso reranker e lo stesso numero massimo di risultati finali, pari a tre.

In questo modo, le differenze osservate tra le pipeline RAG possono essere ricondotte principalmente alle modalità con cui i documenti vengono estratti, strutturati e suddivisi in chunk. Rimangono tuttavia possibili effetti indiretti dovuti al diverso numero di unità prodotte, alla loro lunghezza, alla qualità dei metadati e alla capacità di preservare la continuità delle procedure.

L'approccio black-box è stato valutato utilizzando la stessa ground truth composta dal testo completo della procedura attesa e dalle relative pagine. In questa configurazione, tuttavia, il sistema restituisce direttamente una procedura estratta dal PDF, senza produrre una graduatoria esplicita di chunk. Il

confronto è stato pertanto effettuato applicando metriche compatibili alle pagine e al contenuto restituiti, pur tenendo conto delle differenze architetturali rispetto alle pipeline RAG.

La baseline black-box permette di confrontare una soluzione modulare, nella quale parsing, indicizzazione e retrieval sono osservabili e configurabili, con un approccio end-to-end nel quale l'analisi del documento e la selezione delle informazioni sono delegate al modello multimodale. I risultati devono tuttavia essere interpretati considerando che i due approcci differiscono per grado di controllo, osservabilità, requisiti computazionali e modalità di accesso al documento.

La valutazione della generazione mediante LLM-as-a-Judge è stata invece approfondita sulla configurazione finale della pipeline personalizzata, basata su retrieval ibrido, reranking e materializzazione dei parent, individuata come soluzione principale del lavoro.

Capitolo 8

Analisi dei risultati

Il presente capitolo analizza i risultati ottenuti dalle strategie di elaborazione documentale e dalle configurazioni sperimentali considerate nel lavoro. Il confronto riguarda sia la capacità di individuare le pagine contenenti la procedura attesa, sia il grado di corrispondenza tra il contenuto recuperato o estratto e il testo di riferimento annotato nel dataset.

La Tabella 8.1 riporta i risultati complessivi delle pipeline RAG e dell'approccio multimodale black-box. Le configurazioni RAG comprendono la baseline aziendale, l'approccio basato su Unstructured, la pipeline personalizzata e la strategia che utilizza gli artefatti prodotti da MinerU2.5 per il preprocessing dei documenti. L'approccio black-box elabora invece direttamente il manuale PDF e restituisce la procedura individuata senza esporre una fase separata di indicizzazione e retrieval.

Le metriche con prefisso *P-* descrivono la capacità di localizzare le pagine della procedura di riferimento, mentre quelle con prefisso *R-* valutano la corrispondenza tra il contenuto recuperato o estratto e la ground truth testuale. Le misure di Sequence Similarity, Embedding Similarity e BERTScore F1 forniscono ulteriori indicazioni rispettivamente sull'allineamento sequenziale e sulla vicinanza semantica rispetto alla procedura attesa.

L'analisi viene articolata distinguendo le prestazioni relative al recupero delle informazioni da quelle riguardanti la generazione della risposta. Vengono inoltre discussi i compromessi tra precisione, copertura del contenuto, granularità delle unità documentali, osservabilità della pipeline e costo computazionale. Il capitolo si conclude con l'analisi dei principali limiti emersi dalla sperimentazione.

8.1 Prestazioni del retrieval

Come mostrato nella Tabella 8.1, le strategie analizzate presentano differenze rilevanti sia nella capacità di localizzare le pagine della procedura attesa, sia nella corrispondenza tra il contenuto recuperato o estratto e la ground truth testuale. Nel complesso, la pipeline personalizzata mostra il comportamento più equilibrato, ottenendo i valori migliori in diverse metriche particolarmente rilevanti per il caso applicativo.

Model Type	Models	P-Recall ↑	P-MRR ↑	P-Precision ↑	R-Precision ↑	R-Recall ↑	R-F1 ↑	SeqSim ↑	EmbSim ↑	BERTScore ↑
Pipeline Tools	pipeline iniziale	0.4360	0.6357	0.7429	0.8177	0.7229	0.7181	0.5356	0.9394	0.7780
	unstructured (hi-res)	0.7723	0.9307	0.9238	0.8937	0.6603	0.7036	0.5249	0.9544	0.8120
	pipeline avanzata	**0.9443**	0.9333	0.9381	**0.9475**	**0.9058**	**0.9072**	0.6870	0.9769	**0.9117**
General LLMs	gemini-3.1-flash	0.7572	0.9337	0.9357	0.9454	0.7372	0.7933	0.7057	0.9660	0.8660
	gemini-3.1-pro	0.8888	**0.9548**	**0.9403**	0.9249	0.9041	0.8941	**0.8143**	**0.9822**	0.9080
Specialized VLMs	mineru2.5-1.2b	0.9076	0.9390	0.9012	0.9262	0.8426	0.8600	0.6745	0.9759	0.8460

Tabella 8.1: Confronto complessivo tra le strategie valutate sulle metriche di recupero delle pagine e di aderenza testuale alla procedura di riferimento. Valori più elevati indicano prestazioni migliori.

La baseline aziendale presenta le prestazioni più basse nelle metriche relative alle pagine, con una P-Recall pari a 0.4360, una P-MRR di 0.6357 e una P-Precision di 0.7429. Il valore ridotto di P-Recall indica che, tra i primi tre risultati, anche il miglior intervallo recuperato copre mediamente meno della metà delle pagine annotate per la procedura di riferimento. La P-MRR mostra inoltre che il primo risultato caratterizzato da almeno una pagina pertinente compare meno frequentemente nelle posizioni iniziali rispetto alle altre configurazioni.

Tale comportamento è coerente con le caratteristiche della segmentazione adottata dalla baseline. Il testo viene accumulato fino al riconoscimento di un nuovo titolo o sottotitolo, senza applicare una dimensione massima ai chunk e senza utilizzare una rappresentazione gerarchica. Le sezioni estese possono quindi essere rappresentate da blocchi contenenti argomenti differenti, nei quali la parte pertinente alla domanda incide soltanto parzialmente sulla rappresentazione complessiva.

Sul piano testuale, la baseline raggiunge una R-Precision di 0.8177 e una R-Recall di 0.7229. Quando viene individuato un contenuto pertinente, questo presenta quindi una sovrapposizione non trascurabile con la procedura attesa, ma include elementi estranei oppure non ne copre tutte le parti. La R-F1 pari a 0.7181 sintetizza questo bilanciamento non ottimale tra precisione e copertura.

L'approccio basato su Unstructured migliora sensibilmente la localizzazione documentale. La P-Recall aumenta fino a 0.7723, mentre P-MRR e P-Precision raggiungono rispettivamente 0.9307 e 0.9238. Il primo risultato che condivide almeno una pagina con la ground truth viene pertanto collocato frequentemente nelle prime posizioni e, tra i primi tre risultati, è generalmente presente almeno un intervallo caratterizzato da una buona precisione.

Il miglioramento nelle metriche di pagina non è tuttavia accompagnato da un incremento equivalente della copertura testuale. La R-Precision sale a 0.8937, mentre la R-Recall scende a 0.6603 e la R-F1 si attesta a 0.7036. I risultati suggeriscono che l'approccio individui con buona accuratezza porzioni collocate nell'area corretta del documento, ma che il contenuto selezionato non comprenda sempre l'intera procedura annotata. Questo comportamento può essere associato a confini di sezione non perfettamente coincidenti con quelli della ground truth oppure a errori nella classificazione automatica dei titoli.

La pipeline personalizzata ottiene le migliori prestazioni tra le configurazioni RAG e presenta i valori complessivamente più elevati nelle metriche di copertura. La P-Recall raggiunge 0.9443, indicando che il miglior risultato presente nella top-3 include mediamente quasi tutte le pagine annotate. Anche la P-Precision risulta elevata, con un valore pari a 0.9381, mentre la P-MRR di 0.9333 mostra

che un risultato con almeno una pagina pertinente compare generalmente nelle prime posizioni.

Il vantaggio risulta particolarmente evidente nelle metriche sul contenuto. La pipeline personalizzata raggiunge una R-Precision di 0.9475, una R-Recall di 0.9058 e una R-F1 di 0.9072, ottenendo il valore migliore in tutte e tre le misure. Il contenuto recuperato presenta quindi un'elevata sovrapposizione con la procedura attesa, includendone gran parte degli elementi e limitando al tempo stesso la presenza di testo estraneo. Anche il BERTScore F1, pari a 0.9117, è il più elevato tra le configurazioni considerate e indica un forte allineamento contestuale con la ground truth.

Questi risultati sono coerenti con le scelte introdotte nella pipeline personalizzata. Il filtraggio dei falsi titoli mira a produrre sezioni maggiormente aderenti alla struttura effettiva dei manuali, mentre la rappresentazione parent/child separa l'unità utilizzata per il retrieval da quella fornita come contesto. I child permettono di localizzare porzioni specifiche rispetto alla domanda e la successiva materializzazione dei parent consente di recuperare sezioni più complete. Le metriche aggregate non permettono tuttavia di isolare quantitativamente il contributo di ciascun componente; una valutazione di tipo ablativo sarebbe necessaria per attribuire il miglioramento alle singole scelte progettuali.

Anche la configurazione basata su MinerU2.5 ottiene risultati competitivi, in particolare nella localizzazione delle procedure. La P-Recall è pari a 0.9076 e la P-MRR raggiunge 0.9390, mostrando che la conversione del documento in una rappresentazione strutturata consente generalmente di preservare le informazioni necessarie a individuare le pagine pertinenti.

Le metriche testuali rimangono tuttavia inferiori a quelle della pipeline personalizzata. La R-Recall e la R-F1 sono rispettivamente pari a 0.8426 e 0.8600, mentre il BERTScore F1 si ferma a 0.8460. La differenza può dipendere sia dagli errori introdotti durante l'analisi visuale e la conversione in Markdown, sia dalle successive operazioni di filtraggio, associazione delle pagine e segmentazione. Non è quindi possibile attribuirla esclusivamente al modello MinerU2.5.

Gli approcci black-box basati su Gemini mostrano un comportamento differente. Gemini 3.1 Flash raggiunge una R-Precision elevata, pari a 0.9454, ma una R-Recall più contenuta, pari a 0.7372. Il testo estratto risulta quindi generalmente pertinente, ma non include sempre l'intera procedura di riferimento. La R-F1 pari a 0.7933 riflette tale squilibrio tra selettività e completezza.

Gemini 3.1 Pro ottiene i valori migliori in P-MRR, P-Precision, Sequence Similarity ed Embedding Similarity, rispettivamente pari a 0.9548, 0.9403, 0.8143 e 0.9822. Il modello individua quindi frequentemente una pagina pertinente nelle prime posizioni dell'elenco restituito e produce un contenuto fortemente affine alla ground truth sia sul piano sequenziale sia su quello semantico.

La pipeline personalizzata supera tuttavia Gemini 3.1 Pro in P-Recall, R-Precision, R-Recall, R-F1 e BERTScore F1. La differenza più rilevante riguarda la copertura: la P-Recall e la R-Recall più elevate indicano una maggiore capacità di includere rispettivamente le pagine e il contenuto necessari a ricostruire la procedura completa. Nel dominio considerato, tale proprietà assume particolare importanza, poiché l'omissione di condizioni preliminari, avvertenze o passaggi

operativi può ridurre l'utilità della risposta anche quando il testo restituito è semanticamente affine a quello atteso.

Il confronto tra gli approcci RAG e quelli black-box deve comunque essere interpretato con cautela. Le pipeline RAG restituiscono una graduatoria di unità documentali successivamente materializzate, mentre Gemini produce direttamente una procedura e un elenco ordinato di pagine. Le stesse metriche permettono un confronto rispetto alla ground truth, ma i processi attraverso cui vengono ottenuti i risultati e il significato operativo della loro posizione non sono perfettamente equivalenti.

Nel complesso, i risultati mostrano che gli approcci black-box possono raggiungere valori molto elevati di similarità sequenziale e semantica nell'estrazione diretta delle procedure. La pipeline personalizzata offre tuttavia il miglior equilibrio tra localizzazione delle pagine, precisione e completezza del contenuto, mantenendo inoltre un controllo esplicito sulle fasi di elaborazione documentale, indicizzazione, retrieval e materializzazione.

8.2 Prestazioni della generazione

La qualità della fase generativa è stata valutata sulla configurazione finale della pipeline personalizzata mediante un approccio *LLM-as-a-Judge*. Per ciascuna domanda, il modello valutatore ha confrontato la risposta generata con la procedura di riferimento e con il contesto effettivamente fornito al modello generativo. I punteggi medi ottenuti per correttezza, fedeltà al contesto e completezza sono riportati nella Tabella 8.2.

Criterio	**Punteggio ↑**
Correctness	**4.92/5**
Faithfulness	**4.98/5**
Completeness	**4.69/5**

Tabella 8.2: Valutazione della risposta generata mediante LLM-as-a-Judge.

Il punteggio di *correctness*, pari a 4.92 su 5, indica un'elevata corrispondenza tra le risposte generate, le domande formulate e le procedure di riferimento. Nella maggior parte dei casi, il modello riesce quindi a utilizzare il contesto recuperato per produrre risposte tecnicamente coerenti e pertinenti, preservando il significato delle istruzioni contenute nei manuali.

Il valore più elevato è stato ottenuto per la *faithfulness*, con un punteggio medio di 4.98 su 5. Questo risultato indica che le affermazioni contenute nelle risposte risultano quasi sempre supportate dalle fonti fornite al modello. Il comportamento osservato è coerente con le istruzioni definite nel prompt, che richiedono di non utilizzare conoscenze esterne e di non introdurre passaggi, valori, cause o avvertenze non presenti nel contesto recuperato.

La *completeness* raggiunge un punteggio medio di 4.69 su 5. Sebbene il valore sia elevato, risulta inferiore rispetto a quelli ottenuti per correttezza e fedeltà. In alcuni casi, le risposte possono pertanto omettere dettagli, condizioni

preliminari, note, avvertenze o controlli conclusivi presenti nella procedura di riferimento. Tali omissioni possono dipendere dalla completezza del contesto recuperato, dalla delimitazione delle sezioni documentali oppure dalla richiesta, inserita nel prompt, di produrre risposte concise.

La differenza tra faithfulness e completeness evidenzia un comportamento complessivamente prudente del sistema. Il modello tende a limitarsi alle informazioni chiaramente supportate dal contesto, anche quando ciò comporta l'omissione di alcuni elementi della procedura. Nel dominio applicativo considerato, questo comportamento riduce il rischio di introdurre indicazioni non documentate; tuttavia, anche l'assenza di un passaggio operativo o di un'avvertenza può risultare rilevante e deve pertanto essere considerata un limite.

I risultati evidenziano inoltre la relazione tra la qualità del contesto recuperato e quella della risposta finale. La materializzazione delle unità parent rende disponibili sezioni più ampie rispetto ai frammenti utilizzati durante il retrieval e può contribuire a preservare condizioni, avvertenze e passaggi circostanti. I punteggi aggregati non consentono tuttavia di isolare il contributo specifico della materializzazione rispetto a quello del reranking, del prompt o del modello generativo.

Le risposte possono risultare parzialmente incomplete quando le informazioni necessarie sono distribuite tra sezioni differenti, quando una procedura oltrepassa i confini dell'unità parent oppure quando il retrieval non seleziona tutti i contenuti rilevanti. In questi casi, il modello può formulare correttamente la risposta sulla base del contesto disponibile senza riuscire a coprire integralmente la procedura annotata nella ground truth.

Nel complesso, la configurazione valutata produce risposte caratterizzate da elevata correttezza e forte aderenza alla documentazione. Il principale margine di miglioramento riguarda la copertura di tutti gli elementi rilevanti della procedura, mantenendo al tempo stesso il livello di fedeltà osservato.

8.2.1 Esempio di interrogazione tramite API

Al fine di mostrare il comportamento della pipeline in un caso concreto, è stata eseguita un'interrogazione del sistema mediante l'endpoint `/chat`. L'esempio riportato non ha lo scopo di introdurre una nuova metrica di valutazione, ma di illustrare qualitativamente il flusso che conduce dalla domanda dell'utente alla risposta finale generata dal modello.

La richiesta inviata al sistema contiene il titolo del manuale da interrogare e la domanda formulata in linguaggio naturale. A partire da tali informazioni, la pipeline individua il documento indicizzato, esegue l'eventuale riscrittura della query, recupera le sorgenti più pertinenti e genera una risposta fondata sul contesto restituito dal modulo di retrieval.


```
{
  "document_title": "c32 marine",
  "query": "which are procedure for long-term storage",
  "intent": "procedure for long-term storage",
  "rewritten_queries": [
    "long-term storage procedure",
```

```
    "storage protocol duration",
    "store long term steps"
  ],
  "procedure_extracted": {
    "title": "Engine",
    "content": "Engine\nStorage - Greater than 7 Days and less
    than 30 Days\nBefore placing the engine into storage, clean
     the external surfaces and remove any traces of oil, dust,
    or moisture. Apply a light protective coating to exposed
    metal components to reduce the risk of corrosion during the
     storage period.\n\nKeep the engine in a dry and ventilated
     area. The storage location should be protected from direct
     weather exposure, excessive humidity, and sudden
    temperature changes. \n...\n. If unusual conditions are
    found, stop the engine and correct the issue before
    continuing operation.",
    "page_start": 31,
    "page_end": 34
  },
  "sources": [
    {
      "score": 2.9687533378601074,
      "title": "Engine",
      "content": "Engine\nStorage - Greater than 7 Days and
    less than 30 Days\nBefore placing the engine into storage,
    clean the external surfaces and remove any traces of oil,
    dust, or moisture. Apply a light protective coating to
    exposed metal components to reduce the risk of corrosion
    during the storage period.\n\nKeep the engine in a dry and
    ventilated area. The storage location should be protected
    from direct weather exposure, excessive humidity, and
    sudden temperature changes. \n...\n. If unusual conditions
    are found, stop the engine and correct the issue before
    continuing operation.",
      "page_start": 31,
      "page_end": 34,
      "section": "Towing Information",
      "chapter": "Operation Section"
    },
    {
      "score": 1.232042908668518,
      "title": "Engine Storage Procedure - Check",
      "content": "Engine Storage Procedure - Check\nSMCS Code:
    1000-535\nFor engines that are used intermittently or
    stored for part of the year, the maintenance interval may
    be adjusted when the proper preservation and start-up
    procedures are followed. \n...\n. After the engine has
    reached operating temperature, shut it down and perform a
    final inspection for leaks, abnormal noises, loose
    connections, or warning indicators.",
      "page_start": 109,
      "page_end": 110,
      "section": "Engine Speed/Timing",
```

```
      "chapter": "Maintenance Section"
    }
  ],
  "answer": "For long-term storage exceeding 30 days and less
    than 1 year, the engine should be protected from moisture,
    dust, and temperature changes that may cause condensation
    or corrosion. If the engine remains unused for an extended
    period, lubricant can settle away from internal components,
     leaving surfaces such as cylinder walls, bearings, gears,
    and shafts more vulnerable to oxidation [Engine, 31 - 34].
    \n...\n 16. Attaching a tag showing the storage date and
    inspection schedule to the engine [Engine, 31 - 34].\n17.
    Removing the cover every 2 or 3 months to inspect for
    moisture or corrosion and repeating the protection
    procedure if deterioration is found [Engine, 31 - 34]."
}
```


Listing 8.1: Esempio di risposta completa restituita dall'endpoint `/chat`.

Nell'esempio mostrato la risposta dell'API comprende non soltanto il testo finale generato, ma anche le informazioni intermedie prodotte dalla pipeline. In particolare, sono presenti l'intento associato alla richiesta, le query riscritte, la procedura estratta e l'elenco delle sorgenti utilizzate per la generazione.

8.3 Limiti del sistema

Nonostante i risultati ottenuti, il sistema presenta alcuni limiti legati sia alle scelte progettuali della pipeline sia alla metodologia sperimentale adottata.

Un primo limite riguarda il parsing dei documenti. La pipeline custom utilizza informazioni tipografiche, come la dimensione e lo stile del carattere, insieme a regole definite per riconoscere i titoli e filtrare gli elementi non significativi. Questo approccio ha mostrato buone prestazioni sui manuali analizzati, ma rimane dipendente dalle convenzioni grafiche adottate dai produttori. Documenti con font, impaginazioni o strutture molto differenti potrebbero richiedere una nuova regolazione delle soglie e delle euristiche.

La pipeline opera principalmente sul contenuto testuale del PDF e non interpreta in maniera completa le informazioni contenute nelle immagini, negli schemi elettrici e nelle rappresentazioni grafiche. Tabelle complesse, diagrammi e relazioni espresse esclusivamente attraverso la disposizione visuale possono quindi essere estratti in modo parziale oppure non essere inclusi nel contesto fornito al modello. Questo limite è particolarmente rilevante quando la risposta dipende dalla lettura congiunta di testo e figure.

Anche la strategia parent/child introduce un compromesso. La ricerca sui child favorisce l'individuazione di porzioni specifiche, mentre la materializzazione del parent permette di recuperare il contesto della sezione. Tuttavia, se il parent è molto esteso, il contesto finale può includere informazioni non direttamente pertinenti alla domanda.

La segmentazione dei child dipende inoltre da parametri prestabiliti, come la dimensione massima e la sovrapposizione tra chunk consecutivi. Sebbene tali

valori siano stati scelti in modo da preservare il contesto locale, non esiste una configurazione ottimale per tutti i documenti. Procedure molto brevi, sezioni particolarmente lunghe e contenuti caratterizzati da strutture differenti possono richiedere granularità diverse.

La qualità del retrieval rimane condizionata dai modelli e dai parametri utilizzati. Gli embedding possono non rappresentare correttamente termini molto specialistici. Il query rewriting può aumentare la copertura della ricerca, ma introduce il rischio che una formulazione alternativa modifichi parzialmente l'intento originale. Analogamente, il reranker migliora generalmente l'ordinamento dei candidati, ma comporta un costo computazionale aggiuntivo.

La qualità della risposta generata dipende strettamente dalle informazioni recuperate nella fase di retrieval. Se una parte della procedura non viene inclusa nel contesto fornito al modello, quest'ultimo non è in grado di utilizzarla nella risposta, anche se tale informazione è presente nel manuale originale. Per questo motivo una risposta può risultare corretta e coerente con il contesto disponibile, ma non completamente esaustiva. Questo comportamento è confermato dai risultati della valutazione: il punteggio di completezza, pur essendo elevato, risulta leggermente inferiore rispetto a quelli di correttezza e fedeltà al contesto. In pratica, il sistema privilegia la produzione di informazioni supportate dalle fonti recuperate, riducendo il rischio di allucinazioni, ma può talvolta tralasciare dettagli, note o avvertenze presenti nella documentazione.

Un ulteriore limite riguarda il dataset sperimentale. La valutazione è stata condotta su 70 domande ricavate da otto manuali tecnici. Sebbene i documenti appartengano a categorie differenti, il campione non rappresenta l'intera varietà della manualistica presente nel settore dello yachting. I risultati non possono pertanto essere generalizzati automaticamente a qualsiasi produttore, apparato o tipologia di documento.

Anche la valutazione mediante LLM-as-a-Judge presenta alcuni limiti. I punteggi dipendono dal modello valutatore, dal prompt e dalla scala adottata e potrebbero non coincidere interamente con il giudizio di un esperto tecnico. La valutazione automatica dovrebbe quindi essere affiancata, in un'applicazione reale, da verifiche condotte da personale con esperienza nel dominio.

Capitolo 9

Conclusioni

Il presente lavoro di tesi ha affrontato il problema dell'accesso alla documentazione tecnica nel settore degli yacht di lusso, concentrandosi sulla progettazione e sulla valutazione di una pipeline di Retrieval-Augmented Generation per l'interrogazione di manuali in formato PDF. L'attività, sviluppata nell'ambito del tirocinio svolto presso SailADV, ha riguardato documenti relativi a differenti apparati installati a bordo, caratterizzati da strutture grafiche eterogenee, procedure distribuite su più pagine e frequenti avvertenze di sicurezza.

L'analisi iniziale ha evidenziato come la qualità di un sistema RAG non dipenda esclusivamente dal modello linguistico impiegato, ma dall'intera catena di elaborazione che precede la generazione della risposta. Il parsing dei PDF, il riconoscimento della struttura documentale, la segmentazione del testo, la rappresentazione delle unità informative e la conservazione dei metadati influenzano direttamente la capacità del sistema di individuare contenuti pertinenti e sufficientemente completi.

Per analizzare tale relazione sono state confrontate differenti strategie di preprocessing e chunking. La pipeline aziendale iniziale è stata utilizzata come baseline, mentre le altre configurazioni hanno impiegato Unstructured, gli artefatti prodotti da MinerU2.5 e una pipeline personalizzata sviluppata durante il tirocinio. Sono stati inoltre considerati approcci black-box basati su modelli multimodali della famiglia Gemini, nei quali il documento PDF viene fornito direttamente al modello senza esporre fasi separate di indicizzazione e retrieval.

Il principale contributo progettuale del lavoro è rappresentato dalla pipeline personalizzata. La soluzione utilizza informazioni testuali e tipografiche, insieme alla posizione degli elementi nella pagina e a regole specifiche di filtraggio, per ricostruire la struttura dei manuali e ridurre il riconoscimento errato di avvertenze, riferimenti a figure ed etichette operative come titoli di sezione.

Il contenuto viene organizzato mediante una rappresentazione gerarchica parent/child. Le unità child permettono di effettuare il retrieval su porzioni testuali granulari, mentre il collegamento con le corrispondenti unità parent consente di recuperare sezioni più ampie durante la costruzione del contesto. Tale strategia separa quindi l'unità utilizzata per localizzare l'informazione da quella fornita al modello generativo.

L'architettura sviluppata integra in Qdrant embedding densi e rappresenta-

zioni sparse basate su BM25. La ricerca semantica e quella lessicale vengono combinate mediante Reciprocal Rank Fusion, mentre un modello Cross-Encoder rivaluta e riordina i candidati recuperati. La successiva materializzazione dei parent permette di fornire al modello generativo un contesto più ampio rispetto al singolo frammento selezionato durante la ricerca.

La valutazione sperimentale è stata condotta su un dataset costruito manualmente a partire da otto manuali tecnici e composto da 70 domande associate alle rispettive procedure, ai contenuti di riferimento e alle pagine di provenienza. La metodologia adottata ha consentito di valutare separatamente la localizzazione delle pagine, la corrispondenza tra il contenuto recuperato e la procedura attesa e la qualità della risposta generata.

I risultati mostrano un miglioramento marcato rispetto alla pipeline aziendale iniziale. La pipeline personalizzata ha raggiunto una Page Recall pari a 0.9443, rispetto a 0.4360 della baseline, e una Retrieval F1 pari a 0.9072, rispetto a 0.7181. Ha inoltre ottenuto i valori più elevati di Retrieval Precision, Retrieval Recall e BERTScore F1 tra tutte le configurazioni considerate.

Il confronto con gli approcci black-box ha evidenziato prestazioni particolarmente elevate per Gemini 3.1 Pro, soprattutto in termini di posizione del primo risultato rilevante e similarità sequenziale e semantica con la procedura di riferimento. La pipeline personalizzata ha tuttavia ottenuto valori superiori nella copertura delle pagine e del contenuto atteso. Tale differenza risulta rilevante nel dominio considerato, nel quale l'omissione di una condizione preliminare, di un'avvertenza o di un passaggio operativo può ridurre l'utilità della risposta.

La soluzione personalizzata presenta inoltre il vantaggio di un'architettura modulare e osservabile, nella quale le fasi di elaborazione documentale, indicizzazione, retrieval, reranking e materializzazione possono essere configurate e valutate separatamente. Gli approcci black-box offrono una maggiore semplicità architetturale, ma rendono più difficile individuare l'origine di eventuali errori e intervenire sulle singole fasi del processo.

Anche la valutazione della generazione ha prodotto risultati positivi. La configurazione finale ha ottenuto punteggi medi pari a 4.92 su 5 per la correctness, 4.98 su 5 per la faithfulness e 4.69 su 5 per la completeness. Il valore particolarmente elevato della fedeltà indica che le risposte risultano quasi sempre supportate dal contesto recuperato. Il principale margine di miglioramento riguarda invece la copertura di tutti i dettagli, delle note e delle avvertenze appartenenti alle procedure più articolate.

Nel complesso, i risultati confermano che, nell'applicazione della RAG alla documentazione tecnica, la rappresentazione del documento costituisce un elemento centrale. Una pipeline progettata tenendo conto delle caratteristiche strutturali e terminologiche del dominio può ottenere prestazioni competitive rispetto a modelli multimodali general-purpose, mantenendo al tempo stesso una maggiore tracciabilità delle fonti e un maggiore controllo sul processo di elaborazione.

Il lavoro dimostra pertanto la fattibilità di un sistema capace di facilitare l'accesso alle procedure contenute nei manuali tecnici di bordo mediante domande formulate in linguaggio naturale. La soluzione non sostituisce la documentazione ufficiale né le competenze del personale tecnico, ma rappresenta uno

strumento di supporto alla consultazione, finalizzato a ridurre i tempi di ricerca e a rendere più immediata l'individuazione delle informazioni rilevanti. L'impiego del sistema in contesti operativi deve comunque prevedere la possibilità di verificare le risposte attraverso i riferimenti ai documenti originali.

9.1 Sviluppi futuri

Il lavoro svolto può essere esteso in diverse direzioni, riguardanti sia l'elaborazione documentale sia il retrieval, la valutazione sperimentale e il possibile impiego del sistema in un contesto produttivo.

Un primo sviluppo riguarda il miglioramento della capacità di elaborare documenti caratterizzati da layout differenti rispetto a quelli analizzati. La pipeline personalizzata utilizza infatti regole tipografiche e testuali definite a partire dalle caratteristiche ricorrenti dei manuali disponibili. L'integrazione di tecniche più generali di analisi del layout potrebbe ridurre la necessità di modificare manualmente soglie ed euristiche per ciascuna nuova tipologia di documento, aumentando la robustezza della soluzione rispetto a produttori e convenzioni grafiche differenti.

Un'ulteriore evoluzione consiste nell'includere in modo più completo le informazioni visuali presenti nei manuali. Tabelle, diagrammi, schemi elettrici e immagini annotate possono contenere dati essenziali che non vengono sempre rappresentati correttamente mediante la sola estrazione testuale. L'integrazione di Vision-Language Models potrebbe consentire di interpretare congiuntamente testo, struttura grafica e relazioni spaziali, mantenendo al tempo stesso la modularità della pipeline RAG e il collegamento tra le informazioni recuperate e le pagine originali.

Anche la strategia di chunking potrebbe essere resa adattiva. La dimensione dei child e l'estensione dei parent potrebbero essere determinate in funzione della struttura della sezione, della densità informativa e della tipologia del contenuto, evitando sia la frammentazione delle procedure sia la generazione di contesti eccessivamente estesi. Sarebbe inoltre utile gestire esplicitamente i collegamenti tra contenuti distribuiti in sezioni differenti, recuperando automaticamente prerequisiti, avvertenze, riferimenti interni e procedure correlate.

La fase di retrieval potrebbe essere migliorata mediante modelli di embedding e reranking maggiormente specializzati per la documentazione tecnica e per il dominio marittimo. Un ulteriore sviluppo potrebbe riguardare l'introduzione di criteri dinamici per la selezione del numero di candidati e del contesto finale, adattando la ricerca alla complessità della domanda. Potrebbero inoltre essere sperimentate strategie di decomposizione delle richieste che richiedono informazioni provenienti da più sezioni o da documenti differenti.

Dal punto di vista sperimentale, sarebbe utile condurre uno studio ablativo per misurare il contributo delle singole componenti della pipeline. Il confronto separato tra filtraggio dei titoli, rappresentazione parent/child, retrieval ibrido, query rewriting, reranking e materializzazione permetterebbe di individuare quali elementi incidano maggiormente sulle prestazioni e quali introducano il principale costo computazionale.

Il dataset potrebbe essere ampliato includendo un numero maggiore di manuali, produttori e tipologie di apparati. Potrebbero inoltre essere introdotte domande ambigue, richieste che non trovano risposta nella documentazione, interrogazioni contenenti errori terminologici e casi che richiedono il recupero congiunto di più sezioni. L'inclusione di esempi negativi permetterebbe di valutare in modo più approfondito la capacità del sistema di riconoscere quando il contesto disponibile non è sufficiente a formulare una risposta affidabile.

Un ulteriore sviluppo riguarda la valutazione con utenti reali, quali comandanti, membri dell'equipaggio, manutentori e tecnici di cantiere. Oltre alla correttezza delle risposte, sarebbe possibile misurare il tempo risparmiato nella consultazione dei manuali, l'utilità operativa del sistema, la facilità di verifica delle fonti e il livello di fiducia degli utilizzatori. Il coinvolgimento di esperti del dominio consentirebbe inoltre di valutare l'effettiva rilevanza delle omissioni e degli errori individuati dalle metriche automatiche.

Per l'impiego in produzione sarebbe infine necessario introdurre funzionalità per la gestione delle versioni dei manuali, l'aggiornamento incrementale degli indici, il controllo degli accessi e il monitoraggio delle risposte. Dovrebbero inoltre essere previsti meccanismi per registrare le fonti utilizzate, rilevare eventuali risposte prive di supporto documentale e consentire la segnalazione degli errori da parte degli utenti.

Tali evoluzioni permetterebbero di trasformare il prototipo sviluppato in uno strumento integrato nei processi aziendali, mantenendo i principi di modularità, tracciabilità, verificabilità e collegamento con la documentazione ufficiale.

Bibliografia

[1] *Azienda ospitante: SailADV*. https://sailadv.com/the-group/.

[2] Vaswani et al. «Attention Is All You Need». In: (2017). URL: https://arxiv.org/abs/1706.03762.

[3] Brown et al. «Language Models are Few-Shot Learners». In: (2020). URL: https://arxiv.org/abs/2005.14165.

[4] Ouyang et al. «Training language models to follow instructions with human feedback». In: (2022). URL: https://arxiv.org/abs/2203.02155.

[5] Lewis et al. «Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks». In: (2020). URL: https://arxiv.org/abs/2005.11401.

[6] *Schema generale pipeline RAG*. https://orq.ai/blog/rag-architecture.

[7] Robertson et al. «The Probabilistic Relevance Framework: BM25». In: (2009). URL: https://dl.acm.org/doi/10.1561/1500000019.

[8] Karpukhin et al. «Dense Passage Retrieval for Open-Domain Question Answering». In: (2020). URL: https://arxiv.org/abs/2004.04906.

[9] Robertson et al. «Document management — Portable document format». In: (2020). URL: https://www.iso.org/standard/75839.html.

[10] Clausner et al. «The Significance of Reading Order in Document Recognition and Its Evaluation». In: (2013). URL: https://primaresearch.org/www/assets/papers/ICDAR2013_Clausner_ReadingOrder.pdf.

[11] R. Smith. «An Overview of the Tesseract OCR Engine». In: (2007). URL: https://static.googleusercontent.com/media/research.google.com/it//pubs/archive/33418.pdf.

[12] Kim et al. «OCR-free Document Understanding Transformer». In: (2022). URL: https://arxiv.org/abs/2111.15664.

[13] N. Reimers e I. Gurevych. «Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks». In: (2019). URL: https://arxiv.org/abs/1908.10084.

[14] *Rappresentazione densa*. https://qdrant.tech/course/essentials/day-1/embedding-models/.

[15] Karpukhin et al. «Dense Passage Retrieval for Open-Domain Question Answering». In: (2020). URL: https://arxiv.org/abs/2004.04906.

[16] *Rappresentazione sparse*. https://qdrant.tech/articles/sparse-embeddings-ecommerce-part-1/.

[17] Cormack et al. «Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods». In: (2009). URL: https://dl.acm.org/doi/10.1145/1571941.1572114.

[18] Y. A. Malkov e D. A. Yashunin. «Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs». In: (2020). URL: https://arxiv.org/abs/1603.09320.

[19] *Analisi layout con Unstructured*. https://unstructured.io/blog/how-to-parse-a-pdf-part-1.

[20] B. Wang et al. «MinerU: An Open-Source Solution for Precise Document Content Extraction». In: (2024). URL: https://arxiv.org/abs/2409.18839.

[21] J. Niu et al. «MinerU2.5: A Decoupled Vision-Language Model for Efficient High-Resolution Document Parsing». In: (2025). URL: https://arxiv.org/abs/2509.22186.

[22] *Architettura MinerU2.5*. https://huggingface.co/opendatalab/MinerU2.5-2509-1.2B.

[23] Jiawei Gu et al. «A Survey on LLM-as-a-Judge». In: (2024). URL: https://arxiv.org/abs/2411.15594.

Elenco delle figure

Elenco delle tabelle

Elenco degli Snippet di Codice