

MARCO GRECO

AI ENGINEER & RESEARCHER



CONTATTI

✉ Puoi contattarmi compilando il form presente nella sezione contattami sul sito

📍 Crotone (KR) / Rende (CS), Italia

🌐 <https://www.linkedin.com/in/marcogreco314/>

SKILLS

- LLM / NLP / RAG/ Agentic AI Dev.
- LLM Wiki
- Arize AI Phoenix, Instrumentation, Trace & Span, Observability, Opentelemetry
- CrewAI, Autogen
- Server MCP
- LangGraph, LangChain, LlamaIndex
- Database vettoriale Qdrant
- Gradio, Streamlit, ...
- Knowledge Graph - Based NLP
- Local LLM, Hugging Face, Ollama
- Gestione della vRAM
- Data Preprocessing & Data Analysis
- Docker
- Prompt Engineering
- Sviluppo di algoritmi e modelli di IA
- SQL, Algebra Relazionale
- Problem Solving
- Apache Spark
- Python, Java, C, Bash
- Scikit-Learn, Pandas, Keras, TensorFlow, ...
- Conoscenza degli strumenti di coding agentico (Claude Code, Google Antigravity, Codex, GitHub Copilot, ...)



PROFILO

Sono laureato magistrale in Ingegneria Informatica con specializzazione in Artificial Intelligence & Machine Learning presso l'Università della Calabria. Attualmente sono impegnato in un tirocinio part-time presso l'azienda Convergence Consulting, focalizzato sulla progettazione e sviluppo software per l'automazione di processi CAD, assistita da strumenti di Generative AI e ho recentemente terminato una borsa di ricerca presso l'Università della Calabria, incentrata sullo sviluppo di tecniche avanzate per l'orchestrazione di agenti AI basati su LLM. Ho un forte interesse per il campo del Natural Language Processing (NLP) e per i settori affini dell'intelligenza artificiale. Quindi mi piace essere sempre aggiornato sulle novità di questo settore e mi piace studiare e testare ogni nuova tecnologia recente. Durante questi anni di studio e lavoro ho realizzato diversi progetti alcuni dei quali sono descritti nella sezione "Progetti" del curriculum.



EDUCAZIONE

Scuola Superiore 2013 - 2018
Liceo Scientifico
Filolao - Crotone

Conservatorio di Musica 2013 - 2018
Corso di musica pre-accademico - Pianoforte
Tchaikovsky - Nocera Terinese

Laurea Triennale Set 2018 - Set 2022
Ingegneria Informatica
Università della Calabria - Rende
Metodi Markoviani, Adattivi e Intelligenti per problemi di ottimizzazione

Laurea Magistrale Ott 2022 - Lug 2025
Ingegneria Informatica, Artificial Intelligence & Machine Learning
Università della Calabria - Rende
Progettazione e Sviluppo di un Sistema Multi-Agente di AI Generativa per la ricerca, Comprensione e Revisione di Offerte di Lavoro e Curricula

Ricercatore borsista Set 2025 - Mar 2026
Future Artificial Intelligence Research (FAIR) - spoke 9
Università della Calabria - Rende
"Tecniche avanzate per l'orchestrazione di agenti AI basati su LLM"

LINGUE

- Italiano (Nativo)
- Inglese (Intermedio)

GITHUB

<https://github.com/marcogreco99>



ESPERIENZE LAVORATIVE

Tirocinio part-time - AI Engineer

Convergence Consulting s.r.l.
Bologna/Bangkok (da remoto)

Ago 2025 - Luglio 2026

“Progettazione e sviluppo software per l'automazione di processi CAD, assistita da strumenti di Generative AI”



PROGETTI

Legal AI Assistant — KG-Enhanced LLM per il Diritto Italiano

Marzo 2026

- Sviluppo di un assistente conversazionale per il dominio legale italiano basato su un'architettura originale che combina RAG con Hybrid Search e un LLM potenziato da Knowledge Graph. Il sistema attinge a due fonti di conoscenza: il Codice Civile indicizzato su Qdrant tramite Hybrid Search a tre livelli (dense con multilingual-e5-large, sparse con BM25, re-ranking con ColBERTv2.0), e la giurisprudenza italiana modellata come Knowledge Graph (~4.700 nodi, ~19.600 archi) con NetworkX.
- Per integrare il grafo nel LLM (Qwen3-1.7B frozen), sono stati implementati in PyTorch nativamente una GAT (Graph Attention Network) pre-addestrata con link prediction, un GraphProjector (MLP, GNN dim 128 → LLM dim 2048) e un modulo di Cross-Attention con gating mechanism, addestrati con loss combinata cross-entropy + InfoNCE.
- L'interfaccia Streamlit supporta una modalità RAG classica (Mistral Large via API con function calling / modelli locali via Ollama) e una modalità KG-Enhanced con pipeline completa grafo-LLM.

Multi-Agent Email Assistant — CrewAI + MCP + AgentMail

Dicembre 2025

- Sviluppo di un sistema multi-agente per la gestione automatica della posta elettronica, basato su CrewAI con processo sequenziale. Il sistema integra un MCP Server (agentmail-mcp) come layer di comunicazione con le API di AgentMail, esponendo i tool disponibili agli agenti tramite MCPServerAdapter.
- Il flusso orchestra due agenti specializzati: un Thread Retriever Agent, che recupera i thread di posta in arrivo da una inbox AgentMail, e un Email Sender Agent, che genera autonomamente una risposta contestualmente rilevante e la invia tramite il tool reply_to_message.
- Come LLM backend viene utilizzato Mistral Small 22B eseguito localmente tramite Ollama sulla piattaforma Kaggle, interfacciato a CrewAI via LiteLLM.

Multi-Agent Email Assistant — CrewAI + MCP + Gmail + Google Calendar

In corso di sviluppo

- Evoluzione del progetto precedente verso una pipeline più completa: sostituzione di AgentMail con l'integrazione nativa di Gmail e Google Calendar, aggiunta di uno storage vettoriale su Qdrant per la ricerca semantica sulle email e implementazione di un meccanismo di creazione automatica di eventi calendario estratti dal contenuto dei messaggi.
- Sono stati implementati quattro agenti specializzati con function tools: un Gmail Retrieval Agent che recupera email tramite Gmail API (OAuth2) estraendo headers, body e label; un Vector Storage Agent che genera embeddings e li indicizza su Qdrant; un Calendar Integration Agent che analizza il contenuto delle email, estrae informazioni temporali e crea eventi su Google Calendar API; un Coordinator Agent, con capacità di delega, che supervisiona l'intero workflow.
- Il sistema supporta anche la ricerca semantica sulle email già indicizzate in Qdrant.

Sistema Multi-Agente per l'Analisi di Curricula e Offerte di Lavoro — CrewAI + Qdrant + Gradio

Luglio 2025

- Sviluppo di un sistema multi-agente per l'analisi automatica di curricula e offerte di lavoro. Il sistema legge un CV in PDF, lo confronta semanticamente con un database di 100.000 offerte di lavoro su Qdrant, valuta la compatibilità delle skills e produce un'analisi completa. Tutti gli agenti utilizzano Mistral Large 2411 come LLM e sono orchestrati tramite CrewAI.
- Gli agenti:
 1. PDF Agent — È il punto di ingresso della pipeline. Estrae il testo del CV dal file PDF tramite PDFReadTool (basato su PyPDF2), recupera la categoria professionale di appartenenza del candidato e sintetizza esperienze lavorative e skills. Il suo output diventa il contesto condiviso per tutti gli agenti successivi.
 2. Mathematical Stats Agent — Calcola metriche quantitative sul curriculum. Usa due tool: ScoreCVTool, che conta le skills del candidato e calcola uno score con formula 2^n_{skills} , e TextReadabilityTool, che tramite la libreria textstat calcola un indice di leggibilità del testo del CV.
 3. Job Offers Agent — Recupera le offerte di lavoro più pertinenti da Qdrant. Tramite il QdrantSearchTool, vettorizza la query (concatenazione di categoria e descrizione del CV) e recupera i top 5 risultati più vicini semanticamente. Dal payload di ogni offerta estrae titolo, descrizione e skills richieste.
 4. Soft Skills Agent — Analizza la compatibilità tra le soft skills del candidato e quelle di ciascuna delle cinque offerte recuperate. Senza l'uso di tool, l'LLM distingue soft da hard skills, assegna uno score da 1 a 10 per ogni offerta e fornisce una spiegazione del punteggio.
 5. Hard Skills Agent — Svolge lo stesso ruolo del Soft Skills Agent ma focalizzato sulle hard skills (competenze tecniche), confrontando quelle del CV con quelle richieste da ogni offerta.

- 6. Skill Coherence Evaluator Agent — Calcola la coerenza interna tra soft e hard skills (del CV o delle offerte). Usa il CoherenceTool, che opera in due step: prima calcola la coerenza lessicale tramite un LLM locale (Qwen2.5-1.5B-Instruct), poi calcola la coerenza semantica usando un sentence transformer (all-mpnet-base-v2) con similarità del coseno. Il valore finale è una combinazione lineare dei due.
- 7. Salary Agent — Stima un range salariale adeguato per ogni offerta, tenendo conto degli score di compatibilità calcolati dagli agenti precedenti e del livello di seniority del candidato (junior/senior). Opera interamente tramite ragionamento dell'LLM, senza tool.
- 8. Summary Agent — Agente conclusivo della pipeline sequenziale. Aggrega tutto il contesto prodotto dagli agenti precedenti e genera un riepilogo finale dell'analisi completa, senza l'uso di tool.
- 9. Chatbot Agent — Agente dedicato agli scenari conversazionali. Risponde alle domande dell'utente attingendo alla conoscenza del LLM oppure invocando i tool disponibili (o delegando ad altri agenti nello scenario multi-agent).
- Gli scenari:
 - Scenario Sequenziale — Pipeline automatica senza interazione con l'utente. Gli otto agenti (da PDF Agent a Summary Agent) vengono eseguiti in ordine uno dopo l'altro, ciascuno che alimenta il contesto del successivo. I curricula vengono letti da una struttura di memorizzazione e processati in sequenza. Non c'è nessun input umano durante l'esecuzione.
 - Scenario Conversazionale Single-Agent — Un unico Chatbot Agent esposto tramite interfaccia Gradio. L'utente interagisce in linguaggio naturale e l'agente risponde autonomamente, invocando all'occorrenza tutti i tool disponibili (ricerca su Qdrant, calcolo score, coerenza, leggibilità ecc.). Un solo agente, un solo task, tutti i tool.
 - Scenario Conversazionale Multi-Agent — Scenario più completo. Il Chatbot Agent funge anche da manager: risponde direttamente se ne è capace, oppure delega la richiesta a un agente più specializzato (es. il Job Offers Agent per la ricerca semantica, lo Skill Coherence Evaluator per le coerenze). L'agente specializzato restituisce il risultato al Chatbot Agent, che lo presenta all'utente. Anche questo scenario è accessibile tramite interfaccia Gradio.

Analysis of Social Networks & Media - Bluesky

Aprile 2025

- Il progetto si è focalizzato sull'implementazione e sull'esecuzione di un task di link-prediction nel contesto del social network decentralizzato BlueSky. È stato utilizzato un singolo grafo interazionale composto da 7175 nodi (utenti) e 12309 archi (interazioni, le quali possono essere risposte, ripubblicazioni e citazioni). A seguito di uno studio del grafo sono state analizzate euristiche di link prediction, in particolare, Common Neighbors, Jaccard Coefficient, Adamic Adar, Preferential Attachment e Shortest Path. Inoltre sono stati implementati vari modelli GNN, tra cui GCNConv, SAGEConv e HGTCConv, e utilizzati i modelli LinkPredictor e MLPPredictor.

MathQA: Paper study & Fine-Tuning

Novembre 2024

- Questo progetto è volto allo studio del paper scientifico "*MathQA: Towards Interpretable MathWord Problem Solving with Operation-Based Formalisms*" e ho realizzato un fine-tuning su Mistral 7b, personalizzandolo sul dataset MathQA. In particolare è stata necessaria una formattazione dei prompt per input/output, una quantizzazione a 4-bit (NF4) per ridurre l'uso di memoria, l'utilizzo della libreria PEFT con metodo LoRA per aggiornare solo alcune componenti del modello, un addestramento per oltre 3700 epoche, con checkpoint e salvataggi regolari e test con successiva pubblicazione su Hugging Face Hub

Conversational Agents for Job Recommendation Information Retrieval & NLP

Luglio 2024

- L'obiettivo del progetto è stato quello di recuperare un indice complessivo di compatibilità tra un curriculum vitae e le offerte di lavoro, raccolte attraverso tecniche di scraping da LinkedIn, con l'intento di individuare le posizioni più affini al profilo del candidato.
- Il progetto prevede la creazione di agenti conversazionali utilizzando la libreria Autogen che forniranno un valore di compatibilità complessivo tra il curriculum e le offerte di lavoro selezionate preliminarmente attraverso un calcolo di similarità tra il curriculum e le offerte di lavoro prelevate da LinkedIn con il modello sentence transformer MPNet. Ogni agente è responsabile di un aspetto specifico del curriculum vitae, come Hard Skills, Soft Skills e Salario.
- In assenza di informazioni sul salario, l'agente conversazionale dedicato effettua una stima dello stesso.
- I curriculum vitae utilizzati provengono da un dataset disponibile sul sito di raccolta di dataset Kaggle e hanno richiesto operazioni di preprocessing, inclusi task di Named Entity Recognition (NER) per l'identificazione delle skills.
- La comunicazione tra gli agenti avviene attraverso un Large Language Model (LLM), in particolare Mistral 7B a 4 bit.

Advanced techniques of RAG

Giugno 2024

Methods and Tools for Project Development - ALTILIA

- In un corso universitario, in particolare in collaborazione con alcuni colleghi e sotto la supervisione di esponenti dell'azienda ALTILIA, abbiamo approfondito lo studio della pipeline RAG (Retrieval-Augmented Generation) e implementato tecniche avanzate, tra cui Dense X e RAG Fusion in Python.

Sviluppo di un modello per il riconoscimento e la segmentazione di Transmission Towers (TTs) e Power Lines (PLs).

Giugno 2024

Health Insurance Cross Sell Prediction

Giugno 2023

Data Mining

- Questo progetto rappresenta una delle mie prime esperienze nel campo del data mining e dell'apprendimento automatico, concludendosi con l'addestramento di una semplice rete neurale.
- Il progetto è incentrato su una pratica di vendita nota come *cross-selling*, che consiste nell'offrire un prodotto aggiuntivo a un cliente già acquisito.
- L'obiettivo è quello di prevedere se un cliente di una nota compagnia assicurativa, già titolare di una polizza sanitaria, deciderà di sottoscrivere anche un'assicurazione auto presso la stessa compagnia.

Movie Reviews with Spark

Febbraio 2023

Models and techniques for big data

- In questo progetto ho utilizzato la tecnologia Spark per effettuare query e la struttura dati RDD (Resilient Distributed Dataset) che mi ha permesso di elaborare i dati in modo distribuito.
- Ho deciso di sviluppare un'applicazione web che permetta agli utenti di interagire con i dati estratti dal dataset. A tal fine, è stato utilizzato il framework Flask per la gestione del backend, mentre per il frontend sono stati impiegati HTML e CSS.

Knapsack Problem - ALNS & SA

Settembre 2022

Probability Methods of Operational Research

- Ho realizzato l'implementazione in Python dell'algoritmo Adaptive Neighborhood Search in combinazione della Simulated Annealing su un problema noto della ricerca operativa, il Knapsack Problem.

Altri progetti consultabili sulla mia homepage LinkedIn



CORSI

Evaluating AI Agents

Luglio 2026

DeepLearning.AI - Arize AI