



Costruisci il tuo copilot per i documenti aziendali con Ollama e PostgreSQL

Danilo Dominici





CONSORZIO
UNIVERSITARIO
DI PORDENONE
MOLTIPLICATORE DI VALORE

OVERNET.
upgrade your digital skills

[stesi]
Powered by Innovation

Chi è Danilo Dominici

- Consulente Senior : database... + AI
 - SQL Server
 - PostgreSQL
 - Redis
- Creatore di SQL Start! (sqlstart.it) – evento community a cadenza annuale
- Community speaker, MS Certified Trainer dal 2000, 6x Data Platform MVP
- Email: ddominici@gmail.com



Agenda della presentazione

- Fondamenti di LLM ed embeddings
- Ollama: un LLM engine locale
- PostgreSQL e le estensioni vettoriali
- Architettura di un copilot documentale
- Deployment e scalabilità

Perché un copilot per i documenti aziendali?

- Documenti aziendali crescono in volume e complessità
- Ricerca tradizionale: keyword search limitata
- Rischio: perdita di conoscenza interna
- Soluzione: un copilot AI che risponde in linguaggio naturale

Cos'è un LLM (Large Language Model)?

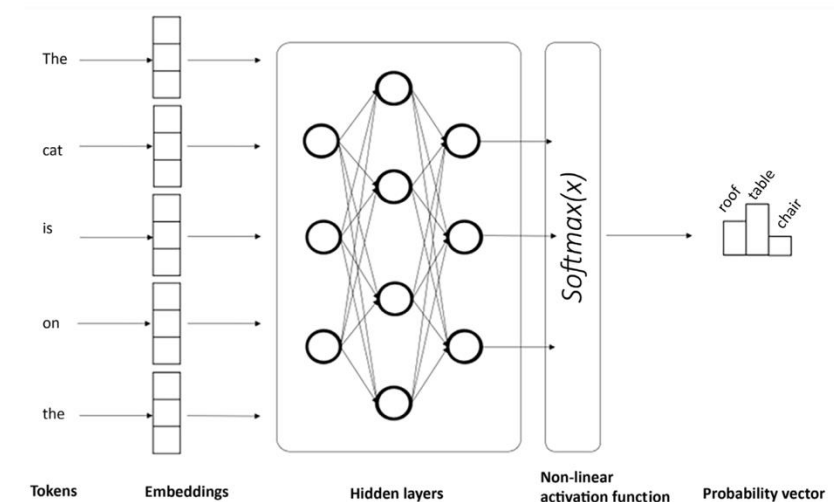
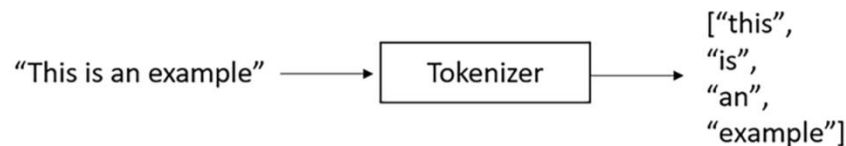
- Modelli di deep learning addestrati su enormi quantità di testo
- Predicono la parola successiva → generano testo coerente
- Esempi: GPT, LLaMA, Mistral
- Usati per chat, Q&A, riassunti, traduzioni

Addestramento e parametri (overview)

- Dati: miliardi di documenti e conversazioni
- Parametri: da miliardi a centinaia di miliardi
- Tecniche: transformer architecture, self-attention
- Risultato: comprensione statistica del linguaggio

Come i LLM comprendono il linguaggio

- Tokenizzazione → trasformano il testo in unità numeriche
- Predizione probabilistica della sequenza di token
- Non *capiscono* semanticamente ma catturano correlazioni
- Output = testo naturale fluido

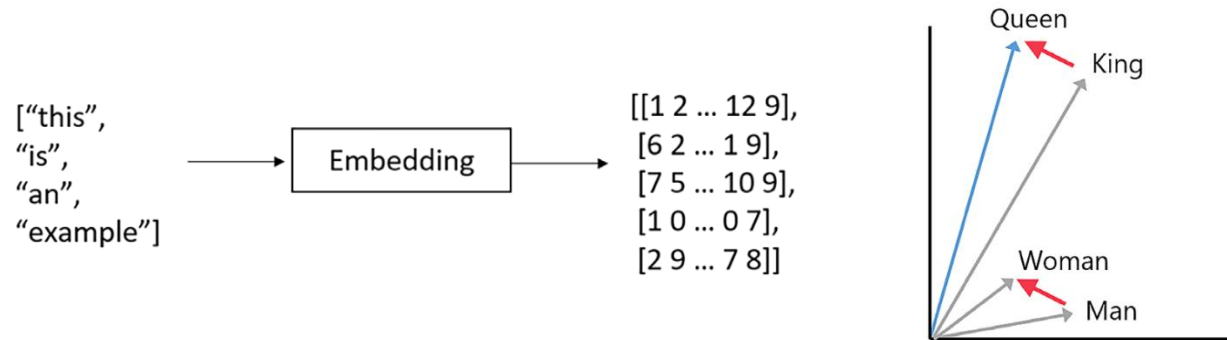


Limiti dei LLM standalone

- Mancanza di aggiornamento → dati 'vecchi'
- Allucinazioni: inventano fatti
- Mancanza di contesto specifico aziendale
- Soluzione: combinarli con embeddings e database

Cosa sono gli embeddings?

- Vettori numerici che rappresentano il significato di un testo
- Frasi simili → vettori vicini nello spazio
- Base per la ricerca semantica



Ricerca semantica con embeddings

- Input utente → embedding → confronto con documenti
- Misura di similarità (coseno, euclidea, ecc.)
- Recupero dei documenti più rilevanti

Pipeline tipica RAG (Retrieval Augmented Generation)

- Query embedding
- Recupero documenti simili
- Inserimento documenti nel prompt LLM
- Generazione risposta con contesto aggiornato

Esempio

REGOLAMENTO AZIENDALE

NORME GENERALI

Le disposizioni che seguono, integrandosi con quanto previsto dal CCNL, intendono enunciare norme e disposizioni aziendali per un corretto rapporto tra l'azienda e personale da questa dipendente.

Si tratta di un codice di comportamento che ha valore generale: Per quanto non espressamente previsto, si rimanda alle procedure interne e alle norme contrattuali e di legge, altrimenti pubblicizzate, nonché a quelle che verranno in futuro emanate.

Ogni dipendente viene a conoscenza del Regolamento all'atto dell'assunzione, ed inoltre questo Regolamento può essere consultato in ogni momento nella bacheca ad esso riservata.

L'inosservanza delle disposizioni e delle norme aziendali costituisce violazione degli obblighi contrattuali e pertanto può dar luogo, anche ove non espressamente previsto e fatta salva ogni eventuale azione legale, a contestazioni formali ed ai conseguenti provvedimenti disciplinari con le procedure previste dal vigente CCNL.

Le disposizioni e le norme contenute nel presente regolamento, nonché le eventuali future ulteriori che ne costituiranno parte integrante, devono essere rispettate non solo dal personale dipendente, ma anche da chiunque intrattenga con l'azienda rapporti di qualunque forma ed a qualunque titolo.

L'inosservanza delle disposizioni e delle norme aziendali da parte di terzi comporterà, fatta salva ogni altra eventuale azione legale, l'allontanamento immediato dai luoghi aziendali.

Tutto il personale riceve disposizioni ed indicazioni di lavoro e richieste di notizie solo ed esclusivamente dai propri superiori diretti.

Utente: 'Qual è la policy ferie 2023?'

Sistema: genera embedding della domanda

Database vettoriale trova documenti HR correlati

LLM produce risposta precisa con citazioni



Ollama

Cos'è Ollama?

- Piattaforma per eseguire modelli LLM in locale
- Semplice installazione, supporto modelli open-source
- Compatibile con Linux, macOS, Windows



Architettura e modelli supportati

- Modelli: LLaMA, Mistral, Falcon, altri
- Ollama = runtime leggero per inferenza
- Accesso via API locale

Cloud

Embedding

Vision

Tools

Thinking

Popular ▾

gpt-oss
OpenAI's open-weight models designed for powerful reasoning, agentic tasks, and versatile developer use cases.

tools thinking cloud 20b 120b

↓ 2.7M Pulls 5 Tags Updated 6 days ago

deepseek-r1
DeepSeek-R1 is a family of open reasoning models with performance approaching that of leading models, such as O3 and Gemini 2.5 Pro.

tools thinking 1.5b 7b 8b 14b 32b 70b 671b

↓ 63.3M Pulls 35 Tags Updated 2 months ago

gemma3
The current, most capable model that runs on a single GPU.

vision 270m 1b 4b 12b 27b

↓ 18.1M Pulls 26 Tags Updated 1 month ago

Vantaggi di Ollama rispetto ad API cloud

- Dati sensibili restano on-premise
- Nessun costo a consumo
- Personalizzazione e tuning
- Funziona offline

Come installare e avviare Ollama

- Installazione con brew o pacchetto dedicato
- Avvio con: `ollama run llama3.2`
- API REST su `localhost:11434`

Esempio pratico

- Comando: `ollama run llama3.2 'Spiegami cos'è un embedding in 3 frasi'`
- Output generato direttamente in locale

Uso con linguaggi di programmazione

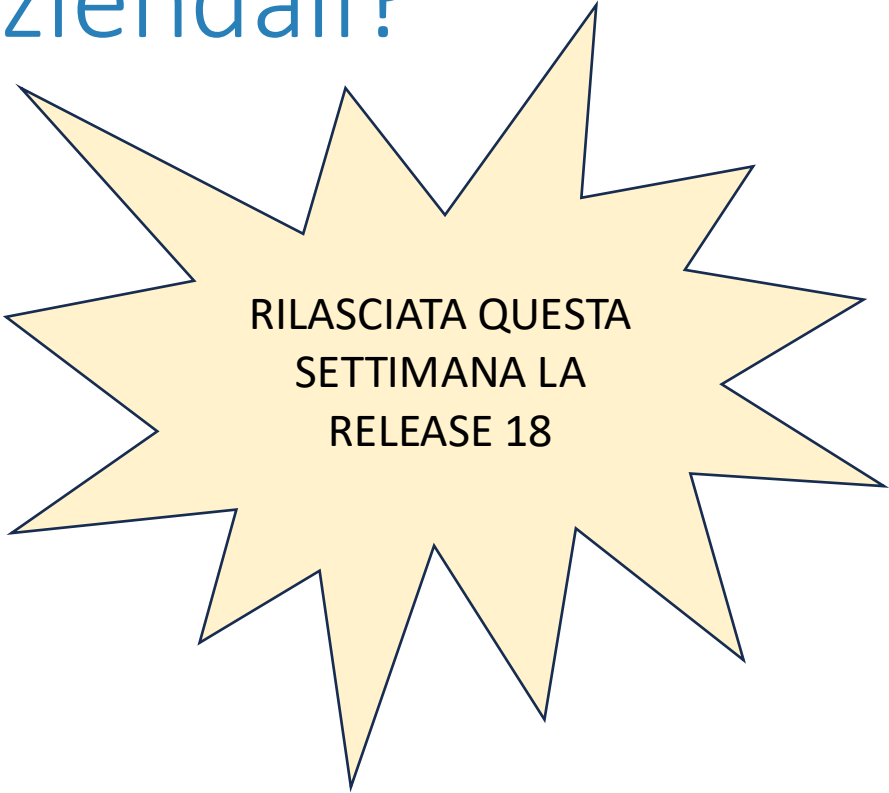
- Python: libreria ollama
- JavaScript/Node.js → client nativo
- Compatibile con LangChain, LlamaIndex



PostgreSQL

Perché PostgreSQL per i dati aziendali?

- Open source, solido, enterprise-ready
- Ampio ecosistema di estensioni
- Ottimo per dati tabellari + vettoriali



RILASCIATA QUESTA
SETTIMANA LA
RELEASE 18

Sfida: embeddings su DB tradizionale

- Ogni embedding = centinaia di dimensioni
- SQL classico non adatto a similarità vettoriale
- Necessarie estensioni specializzate

Estensione pgvector – introduzione

- Nuovo tipo di dato: vector
- Permette salvataggio e query su embeddings
- Standard de facto AI + PostgreSQL

Creare tabelle con colonne vettoriali

```
CREATE TABLE documents (  
    id serial,  
    content text,  
    embedding vector(768)  
);
```

```
INSERT INTO documents (content, embedding)  
VALUES ('Fattura 123 Lucerne Publishing', '[0.12, 0.54, ...]');
```

Query di similarità

SELECT content

FROM documents

ORDER BY embedding <-> '[...]'

LIMIT 5;

<-> = distanza euclidea, supporto cosine e dot product

Indexing vettoriale

- Migliora prestazioni su milioni di embeddings
- Trade-off precisione vs velocità
- Ricerca semantica istantanea

Estensioni complementari

- timescale → dati temporali
- pgai → logica AI in SQL

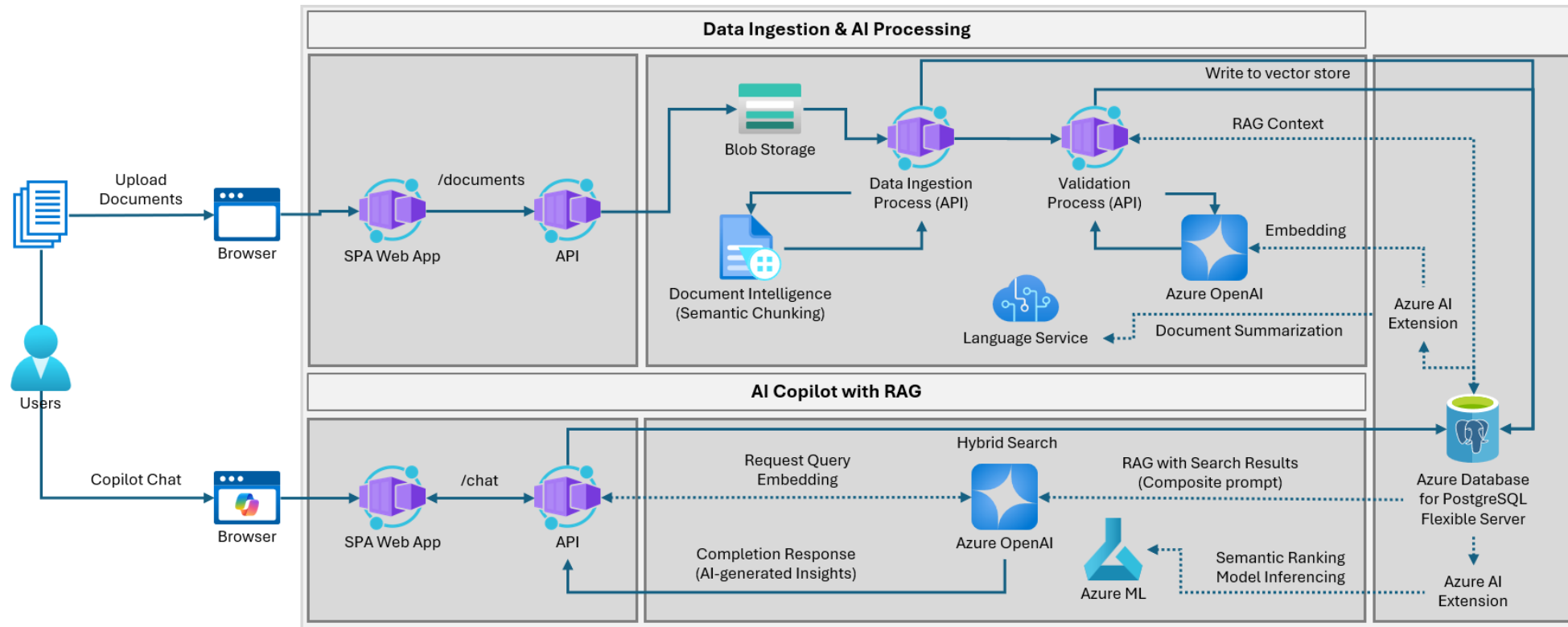
Best practice scalabilità

- Normalizzare embeddings
- Creare indici appropriati
- Evitare full scan
- Monitorare CPU e memoria



Copilot

Architettura originale



<https://solliancenet.github.io/microsoft-postgresql-solution-accelerator-build-your-own-ai-copilot/01-Introduction/>

Architettura "locale"

- Sostituita la componente Azure (OpenAI e Document Intelligence) con Ollama
- Sostituito Azure database for PostgreSQL con PostgreSQL 17 locale + estensione pgvector



DEMO

Esecuzione: locale vs server

- Locale = prototipi
- Server on-premise = enterprise
- Cloud privato = team distribuiti

Containerizzazione con Docker

- Immagini preconfigurate per Ollama e Postgres
- Deploy rapido su Kubernetes
- Scalabilità orizzontale

Monitoraggio prestazioni

- Metriche: latenza query, consumo RAM/CPU
- Strumenti: Prometheus + Grafana
- Logging centralizzato

Scalabilità

- Posso fare lo sharding degli embeddings su PostgreSQL diversi
- Load balancing richieste Ollama
 - Web service: posso usare load balancer hw o sw
- Cache risultati frequenti
 - Redis

Possibili evoluzioni

- Multimodalità: testo + immagini + audio
- Integrazione con CRM, ERP
- Agenti AI autonomi

Cosa mi porto a casa

- LLM potenti ma limitati senza memoria esterna
- Richiesto hardware adeguato, soprattutto con GPU di buona potenza
- PostgreSQL + pgvector = knowledge base semantica
- Ollama = privacy, costo zero, locale
- Copilot documentale = produttività e accesso rapido

Risorse

- Video di Andrej Karpathy
<https://www.youtube.com/andrejkarpathy>
- DeepLearning.AI (Andrew Ng)
<https://www.deeplearning.ai/courses/>
- Video di Dave Ebbelaar
<https://www.youtube.com/@daveebbelaar>
- Newsletters:
 - Stefano Gatti (e newsletters suggerite da lui 😊)
<https://stefanogatti.substack.com>

Risorse

- Progetto originale (Copilot + Azure PostgreSQL)

<https://solliancenet.github.io/microsoft-postgresql-solution-accelerator-build-your-own-ai-copilot/>

- Demo e slides

<https://github.com/ddominici/presentations>

The end

- Q & A
- Contatto : ddominici@gmail.com