t-defence

Retrieval Augmented Generation (RAG) per la consultazione locale e sicura di documenti:

indicizzazione a grafo e retrieval ibrido per la modalità “FAT RAG”

#TDefenceBusiness

AI4Cyber

Summary

Abstract 04

Introduzione 05

Architettura del Sistema 06

Graph Indexing Module 07

Registro delle entità e deduplicazione 08

ER Graph Indexing Module 10

Graph Retrieval Module 11

La modalità FAT RAG 12

Visualizzazione ed esplorazione del grafo 13

Modelli utilizzati 14

Sviluppi futuri 16

Bibliografia 17

AI4Cyber

AI4Cyber studio è il nuovo spazio di ricerca applicata di T-Defence, dedicato alla ricerca e applicazione dell'**Intelligenza Artificiale in ambito cybersecurity**.

Al suo interno, diversi specialisti con competenze ed esperienze eterogenee collaborano per affrontare le sfide poste dal panorama delle minacce informatiche, accompagnare i clienti nell'adozione dell'AI nei propri processi aziendali e condividere le attività di ricerca con la comunità tecnico-scientifica.

Il team cura l'**intero ciclo di sviluppo dei sistemi basati su AI**: dalla raccolta e dal preprocessing dei dati, all'addestramento e validazione dei modelli, fino al deployment in produzione.

Le soluzioni proposte si fondano su tecnologie avanzate di **Machine Learning**, **Deep Learning** e **Large Language Models**, e possono essere personalizzate in base alle specifiche esigenze del cliente.

L'AI Team è costantemente impegnato in attività di ricerca e sperimentazione, con un'attenzione particolare agli **aspetti etici**, alla **trasparenza** e alla **tutela della privacy**.

L'obiettivo è proporre soluzioni innovative che siano in linea con i principi di **equità**, **inclusività** e **rispetto dei diritti fondamentali**.

Abstract

Negli ultimi anni, i sistemi basati su Retrieval-Augmented Generation (RAG) si sono affermati come una delle soluzioni più efficaci per la consultazione di documenti in linguaggio naturale, permettendo di combinare la capacità di ragionamento dei Large Language Models (LLM) con informazioni provenienti da fonti esterne verificabili.

Tuttavia, gli approcci di retrieval basati esclusivamente su rappresentazioni vettoriali dei contenuti indicizzati, pur garantendo delle buone prestazioni di ricerca, tendono a trattare i documenti come insiemi di frammenti indipendenti, perdendo quindi le relazioni concettuali che legano tra loro sezioni, immagini, tabelle e argomenti trattati.

Il presente studio propone un'estensione del sistema già descritto precedentemente in [1] [2] [3], attraverso l'introduzione dell'indicizzazione a grafo. I documenti di interesse dell'utente, dopo la fase di indicizzazione, vengono analizzati da un LLM che ne estrae i concetti chiave, le entità e le relazioni che le collegano, costruendo un grafo di conoscenza contenente la struttura del documento (paragrafi, immagini, tabelle) e il suo contenuto semantico.

Su questa rappresentazione si fonda una nuova modalità di retrieval, denominata "FAT RAG", che combina diverse strategie complementari: la ricerca vettoriale tradizionale, la ricerca lessicale e l'esplorazione del grafo concettuale.

Autori:

- Simone Parrella: AI Security Engineer
- Simona Sorgente: AI Officer

Introduzione

L’architettura RAG tradizionale si basa sulla divisione del documento in frammenti (chunk); ognuno di questi viene trasformato in un vettore numerico tramite un modello di embedding e in maniera analoga, durante l’interrogazione, la domanda dell’utente viene anch’essa trasformata in vettore e confrontata con tutti i frammenti precedentemente ottenuti, allo scopo di restituire quelli più simili.

Questo approccio funziona bene solo nel momento in cui l’informazione è contenuta in un singolo chunk, mostrando limitazioni in scenari realistici; infatti, se la risposta a una domanda è distribuita in più parti del documento o il concetto cercato è espresso con termini diversi da quelli usati nella domanda, la loro similarità vettoriale rischia di restituire un contesto incompleto.

I grafi di conoscenza offrono una risposta a questo problema, rappresentando esplicitamente i concetti presenti nei documenti e le relazioni che li collegano. Questo permette di "navigare" l'informazione anziché limitarsi a cercarla per somiglianza. Quindi, un'interrogazione che parte da un concetto può così raggiungerne altri collegati, i relativi paragrafi che li descrivono e i contenuti multimediali che li illustrano.

Negli ultimi anni si è diffuso il termine “GraphRAG” per indicare proprio il paradigma in cui gli LLM vengono impiegati non solo per generare le risposte, ma anche per costruire automaticamente il grafo a partire dai documenti, estraendo concetti, entità e relazioni senza intervento manuale.

Il presente studio, dunque, descrive come questo paradigma sia stato integrato all'interno del sistema esistente, mantenendo i vincoli che caratterizzano l'intero progetto: esecuzione completamente locale, utilizzo di modelli open source e attenzione alla natura sensibile dei dati trattati.

Architettura del sistema

Il modulo si inserisce all'interno della pipeline di elaborazione documentale già esistente [1], sfruttandone i meccanismi di acquisizione ed estrazione dei contenuti già consolidati. I documenti caricati vengono elaborati dall'Extractor Module, che ne restituisce i frammenti testuali organizzati per sezione, le immagini codificate e le tabelle in formato HTML.

Rispetto alla versione precedente, si introducono tre componenti principali:

- Graph Indexing Module: costruisce il grafo concettuale dei documenti, estraendo tramite LLM i concetti chiave e i sotto-concetti da testo, tabelle e immagini.
- ER Graph Indexing Module: costruisce un secondo grafo, orientato alle entità e alle relazioni, in cui tra due elementi possono coesistere relazioni multiple.
- Graph Retrieval Module: sfrutta il grafo concettuale in fase di interrogazione, combinandolo con l'indice vettoriale esistente nella nuova modalità di retrieval "FAT RAG."

I grafi generati vengono salvati nel database centralizzato tramite GridFS [9] e rimangono associati al singolo utente. L'indicizzazione è incrementale: quando vengono caricati nuovi documenti, il sistema individua automaticamente quelli non ancora presenti nel grafo e li aggiunge senza dover ricostruire l'intera struttura. Inoltre, i concetti comuni a più documenti vengono collegati fra loro, superando i confini del singolo documento e rendendo possibili relazioni cross-documento.

Poiché la costruzione del grafo richiede diverse interazioni con il modello linguistico, il processo è stato organizzato in tre fasi:

- una fase di pianificazione, in cui vengono preparati tutti i compiti di estrazione;
- una fase di esecuzione, in cui le chiamate al modello vengono eseguite in parallelo secondo il numero di worker configurato;
- una fase di riduzione controllata, in cui i risultati vengono integrati nel grafo evitando conflitti di scrittura.

Graph Indexing Module

Il Graph Indexing Module trasforma i metadati estratti dai documenti in un grafo diretto in cui sono presenti due livelli di informazione: la struttura del documento e il suo contenuto semantico.

Per ogni documento viene creato un nodo DOCUMENT, collegato ai nodi PARAGRAPH che rappresentano le sezioni del testo. I paragrafi consecutivi sono collegati tra loro da archi di tipo NEXT, che preservano l'ordine di lettura del documento, le immagini e le tabelle diventano rispettivamente nodi IMAGE e TABLE per poi essere agganciati al paragrafo in cui compaiono.

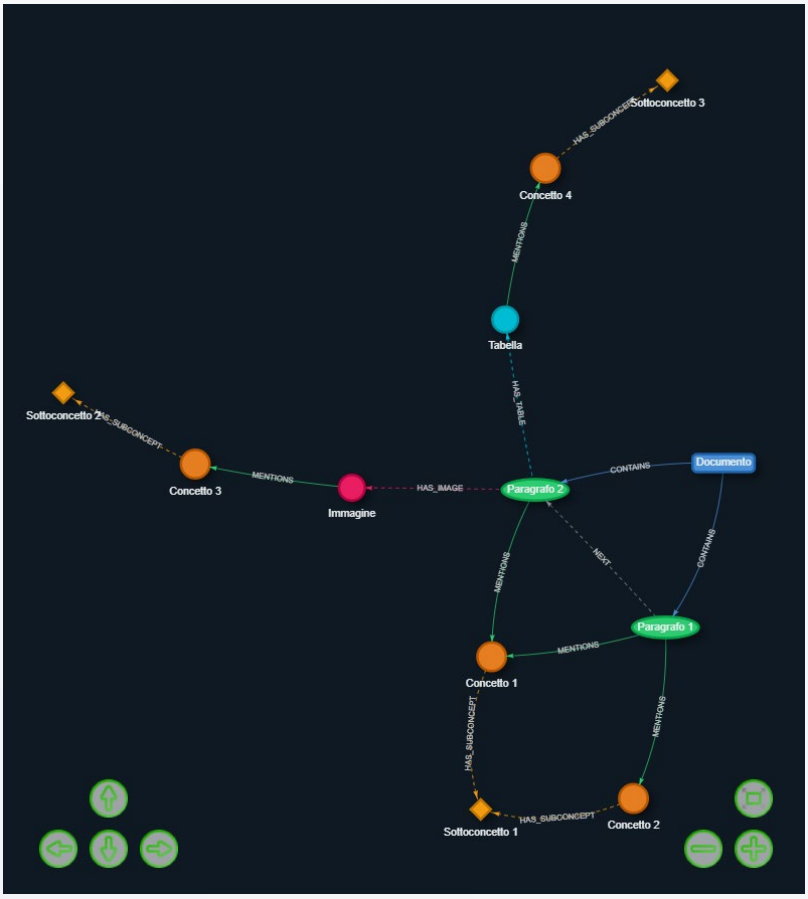

Figura 1: Esempio di grafo che mostra due paragrafi consecutivi con un concetto e un sotto-concetto condivisi, e un'immagine e una tabella da cui emergono a loro volta concetti e sotto-concetti.

Successivamente ogni sezione del documento viene sottoposta all'LLM con un prompt dedicato che chiede di identificare i concetti chiave del testo: componenti, informazioni o elementi strutturali che risultano rilevanti per la comprensione del contenuto. Il prompt definisce con precisione cosa è un concetto, ovvero un'unità specifica del contenuto, effettivamente descritto o caratterizzato nel testo, e cosa non lo è, come termini generici, riferimenti marginali o informazioni dedotte ma non esplicitamente presenti. Per ogni concetto il modello restituisce una breve descrizione, gli eventuali sotto-concetti e le relazioni con altri concetti, il tutto in formato JSON strutturato.

Un aspetto delicato riguarda la modalità con cui si gestisce il testo da sottoporre al modello. Se una sezione testuale è sufficientemente breve da rientrare nella finestra di contesto disponibile viene elaborata per intero, altrimenti viene suddivisa in finestre scorrevoli in cui ogni frammento viene accompagnato da quello successivo per preservare la continuità del discorso; per sezioni molto lunghe è previsto un meccanismo di raggruppamento che limita il numero massimo di finestre, mantenendo i tempi di elaborazione ragionevoli.

Le tabelle seguono un percorso dedicato: il contenuto HTML viene ripulito dai tag e analizzato con un prompt specifico, che chiede di identificare i concetti presenti nelle intestazioni di colonna e nei contenuti delle celle.

Le immagini, infine, vengono elaborate da un Vision Language Model (VLM), che produce prima una descrizione sintetica di ciò che l'immagine mostra e poi l'elenco dei concetti visibili o rappresentati.

Il risultato è un grafo in cui ogni concetto è collegato ai paragrafi, alle immagini e alle tabelle che lo menzionano, ai suoi sotto-concetti e agli altri concetti con cui intrattiene relazioni, con archi pesati che riflettono la forza e la frequenza di tali collegamenti.

Registro delle entità e deduplicazione

Uno dei problemi più noti nella costruzione di grafi di conoscenza è costituito dai duplicati; infatti, lo stesso concetto può comparire nel testo con grafie diverse, con o senza trattini, al singolare o al plurale, in forma estesa o abbreviata. Senza un meccanismo di riconciliazione, il grafo si riempirebbe di nodi che rappresentano la stessa cosa, frammentando l'informazione e compromettendo la qualità del retrieval.

Il sistema affronta il problema su più livelli. In primo luogo, i nodi vengono normalizzati: ogni nome viene ricondotto a una forma canonica, rimuovendo numerazioni di sezione, separando eventualmente le parole scritte in notazione camelCase, eliminando punteggiatura e caratteri speciali e uniformando gli spazi. In secondo luogo, viene utilizzato un registro delle entità (Entity Registry): una struttura centralizzata che tiene traccia di tutti i concetti già presenti nel grafo. Quando l'estrazione produce un nuovo concetto, il registro verifica se esiste già un concetto equivalente applicando tre criteri in cascata:

- L'uguaglianza esatta delle forme normalizzate;
- L'inclusione dei termini (per cui un nome più corto i cui token sono tutti contenuti in un nome più lungo viene considerato una variante dello stesso concetto);
- la similarità fuzzy, calcolata confrontando le stringhe con le parole riordinate alfabeticamente, così da riconoscere come equivalenti anche espressioni con ordine delle parole diverso.

Quando viene trovata una corrispondenza, il nuovo riferimento viene fuso nel nodo esistente, mantenendo come etichetta la denominazione più descrittiva tra quelle presenti. Ad esempio, se uno stesso elemento viene indicato nel testo come “modello linguistico” e successivamente come “modello linguistico di grandi dimensioni”, viene mantenuta quest’ultima denominazione.

A valle della costruzione del grafo è prevista una fase di deduplicazione globale, che riesamina l'intero grafo alla ricerca di coppie di nodi sfuggite al controllo incrementale: i nodi vengono ordinati per frequenza di menzione e i duplicati vengono fusi nel nodo più rappresentativo, trasferendo su di esso tutti gli archi e sommando i pesi.

Un ultimo raffinamento riguarda i sotto-concetti: un elemento estratto come sotto-concetto di un concetto padre può in realtà coincidere con un nodo autonomo già presente nel grafo. Per i casi di corrispondenza esatta la fusione avviene automaticamente, per i casi ambigui, in cui la similarità è alta ma non totale, il sistema interpella un modello linguistico leggero con una domanda binaria, chiedendo se i due elementi rappresentino effettivamente la stessa cosa. La disambiguazione tramite LLM permette di risolvere casi che nessuna regola sintattica potrebbe gestire, mantenendo al tempo stesso un costo computazionale contenuto.

ER Graph Indexing Module

Oltre al grafo concettuale, è possibile generare un secondo grafo entità-relazioni rispondendo alla domanda “chi fa cosa?” piuttosto che “di cosa parla questo documento?”. Il modello in questo caso, tramite un prompt ad hoc, identifica tutte le entità contenute nel testo (persone, luoghi, tecnologie e standard) e le relative relazioni che le collegano (relazioni implicite, gerarchiche e funzionali).

In questo caso c’è una differenza sostanziale rispetto al grafo concettuale descritto in precedenza, grazie all’adozione di un multi-grafo diretto in modo da poter rappresentare più relazioni tra le stesse entità, ognuna con la propria etichetta, descrizione e peso.

Tale scelta riflette una proprietà fondamentale del mondo reale, in cui le entità sono spesso legate da più relazioni contemporaneamente. Nei grafi semplici, invece, tra due nodi può esistere un solo arco e ogni nuova relazione sovrascriverebbe la precedente, perdendo informazione.

L'estrazione delle entità e delle relazioni oltre a seguire il flusso a tre fasi già descritto (pianificazione, esecuzione parallela, riduzione), riutilizza il registro delle entità per la riconciliazione dei duplicati e gestisce allo stesso modo testo, tabelle e immagini. La direzione delle relazioni viene inoltre dettata dal flusso naturale dell'azione, dal soggetto che la compie verso l'oggetto che la subisce, così da rendere il grafo leggibile e interrogabile in modo coerente.

È importante precisare che, nella versione attuale del sistema, il grafo entità-relazioni e il grafo concettuale vengono costruiti e mantenuti separatamente e svolgono funzioni differenti. Il grafo concettuale è quello utilizzato dal Graph Retrieval Module per le operazioni di recupero, mentre il grafo entità-relazioni è attualmente impiegato per la visualizzazione e per analisi esterne sul contenuto estratto.

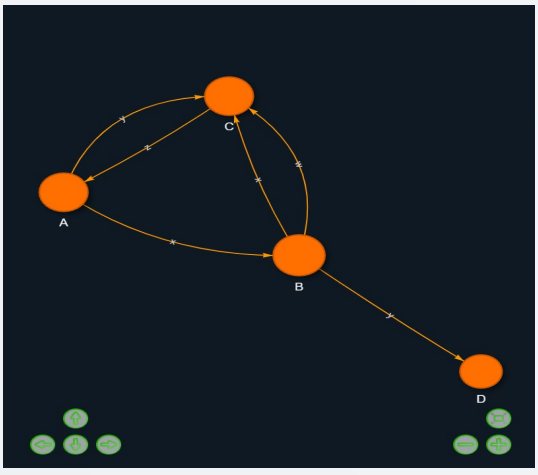

Figura 2: Esempio di grafo entità-relazioni, con nodi generici (A-D) collegati da relazioni multiple, anche parallele tra la stessa coppia di entità.

Graph Retrieval Module

Il Graph Retrieval Module è il componente che mette a frutto il grafo concettuale durante la fase di interrogazione. Il suo funzionamento si articola nei seguenti step:

- Identificazione dei punti di ingresso;
- Attraversamento del grafo;
- Assemblaggio del contesto;
- Riordinamento finale.

Il primo passo consiste nell'individuare i concetti del grafo più pertinenti rispetto alla domanda dell'utente, trasformata anch'essa in concetti e sotto-concetti. Poi, per ogni nodo dei documenti indicizzati viene calcolato un embedding a partire dall'etichetta e dalla descrizione, mantenuto in cache per le interrogazioni successive; la domanda viene confrontata con questi embedding tramite similarità semantica, integrata da un confronto fuzzy tra le stringhe normalizzate che premia le corrispondenze letterali. I concetti che ottengono uno score di corrispondenza pari o superiore a 0,78 vengono selezionati come punti di ingresso "locali", con un peso iniziale unitario, consentendo un'esplorazione del grafo fino a 2 hop; quelli con uno score compreso tra 0,62 e 0,78 vengono invece utilizzati come punti di ingresso globali, con un'esplorazione limitata a 1 hop e un peso iniziale ridotto. Lo score deriva dalla similarità semantica tra la domanda e l'embedding del concetto e può essere incrementato in presenza di una forte corrispondenza fuzzy tra i termini normalizzati. Se nessun concetto raggiunge la soglia minima di 0,62, viene utilizzato un meccanismo di fallback basato sulla sovrapposizione dei termini, che consente comunque di individuare un punto di partenza.

Il secondo passo riguarda l'attraversamento del grafo vero e proprio, realizzato come un'esplorazione in ampiezza. Partendo dai punti di ingresso, il sistema risale dai concetti ai paragrafi che li menzionano, accumulando su ciascun paragrafo un punteggio proporzionale alla rilevanza dei concetti raggiunti. L'esplorazione si propaga anche lungo le relazioni tra concetti, con un decadimento a ogni salto e una penalizzazione per i nodi molto connessi, che essendo collegati a tutto rischierebbero di portare l'esplorazione fuori strada. Alcuni accorgimenti arricchiscono il punteggio: i paragrafi adiacenti a un paragrafo molto rilevante ricevono un incremento del punteggio, sfruttando gli archi NEXT, poiché è probabile che il discorso prosegua nella sezione successiva; le immagini e le tabelle raggiunte durante la visita non compaiono come risultati autonomi, ma trasferiscono un bonus al paragrafo che le contiene.

Quest'ultimo punto ha una conseguenza importante sul terzo passo: quando un paragrafo viene selezionato, il suo contenuto viene ricostruito integrando il testo con le descrizioni delle immagini e il contenuto delle tabelle che gli appartengono. Il modello che genera la risposta riceve quindi un contesto completo, in cui l'informazione visiva e tabellare è resa disponibile in forma testuale accanto al paragrafo di riferimento.

Il punteggio dei paragrafi viene infine arricchito con un contributo derivato dal PageRank, un algoritmo che assegna a ciascun nodo del grafo un valore di centralità in funzione della sua posizione e delle connessioni con gli altri nodi. Nel sistema, il PageRank viene calcolato sull'intero grafo e utilizzato per valorizzare i paragrafi collegati a concetti maggiormente centrali nella struttura del grafo. I risultati vengono sottoposti a un riordinamento finale basato sulla similarità diretta tra la domanda e il contenuto assemblato, che corregge eventuali distorsioni introdotte dall'attraversamento.

La modalità FAT RAG

La modalità FAT RAG rappresenta la sintesi degli elementi illustrati fino ad ora, utilizzando una strategia di retrieval che fonde due canali di ricerca complementari sui documenti.

Il primo canale è la ricerca vettoriale tramite un indice FAISS, già presente nel sistema fin dalle prime versioni [1], che eccelle nel trovare frammenti semanticamente simili alla domanda. Il secondo è il graph retrieval appena descritto, che aggiunge la capacità di seguire le connessioni concettuali e di recuperare contesti distribuiti nel documento.

La fusione tra i due canali avviene a partire dai punteggi prodotti dai rispettivi meccanismi di recupero, dove FAISS assegna a ciascun risultato un punteggio derivato dalla similarità vettoriale tra la domanda e il relativo chunk, mentre il recupero delle informazioni basato su grafo assegna un punteggio ai paragrafi sia in funzione della rilevanza dei concetti raggiunti durante l'attraversamento che del contributo del PageRank. I punteggi dei due canali vengono quindi normalizzati separatamente rispetto al valore massimo ottenuto nel rispettivo insieme di risultati e combinati tramite una media pesata, gestita dal parametro alpha. Con alpha pari a zero il sistema si comporta come un RAG basato su ricerca vettoriale, con alpha pari a uno si utilizza il grafo delle conoscenze, mentre i valori intermedi bilanciano i due contributi.

Di default si attribuisce la stessa importanza ad entrambe le componenti. Inoltre, i risultati provenienti dai due canali vengono combinati sulla base della coppia documento-sezione, così che lo stesso contenuto trovato da più modalità riceva un punteggio combinato anziché comparire più volte.

La modalità supporta inoltre il filtraggio per documento per poter restringere la ricerca a un sottoinsieme del corpus documentale; in questo caso il sistema recupera un numero maggiore di risultati prima di applicare il filtro, in modo da garantire risultati sufficienti anche dopo la riduzione.

Un aspetto progettuale rilevante è la robustezza: se un utente non ha ancora grafi di conoscenza, il sistema ripiega automaticamente sul retrieval tradizionale senza interruzioni del servizio. Il passaggio alla modalità FAT RAG è quindi un potenziamento incrementale, che non introduce nuovi requisiti per l'utente finale.

Visualizzazione ed esplorazione del grafo

Per rendere i grafi consultabili è stato sviluppato un modulo di visualizzazione integrato nell'interfaccia del sistema.

Il modulo offre una vista interattiva del grafo, con la possibilità di filtrare per documento e per tipologia di nodo, di limitare il numero di elementi visualizzati e di scegliere tra un motore di rendering interattivo e uno statico. Inoltre, una sezione statistica riassume la composizione del grafo: numero di nodi e archi per tipo, entità più menzionate, relazioni più frequenti, coppie di entità collegate da più relazioni distinte e copertura delle descrizioni generate per le immagini.

La funzione di ispezione permette di cercare un nodo per parola chiave e di esaminarlo nel dettaglio: per un paragrafo viene mostrato il contenuto testuale, per un'immagine l'anteprima e la descrizione generata dal modello visivo, per una tabella il rendering HTML originale, per un'entità la descrizione, il numero di menzioni, il frammento di testo da cui è stata estratta e l'elenco completo delle relazioni entranti e uscenti.

Il grafo può infine essere esportato nei formati standard GraphML [10], compatibile con strumenti di analisi come Gephi [11] e Cytoscape [12], e CSV, contenente liste di nodi, archi e relazioni, così da consentire analisi esterne o integrazioni con altri sistemi.

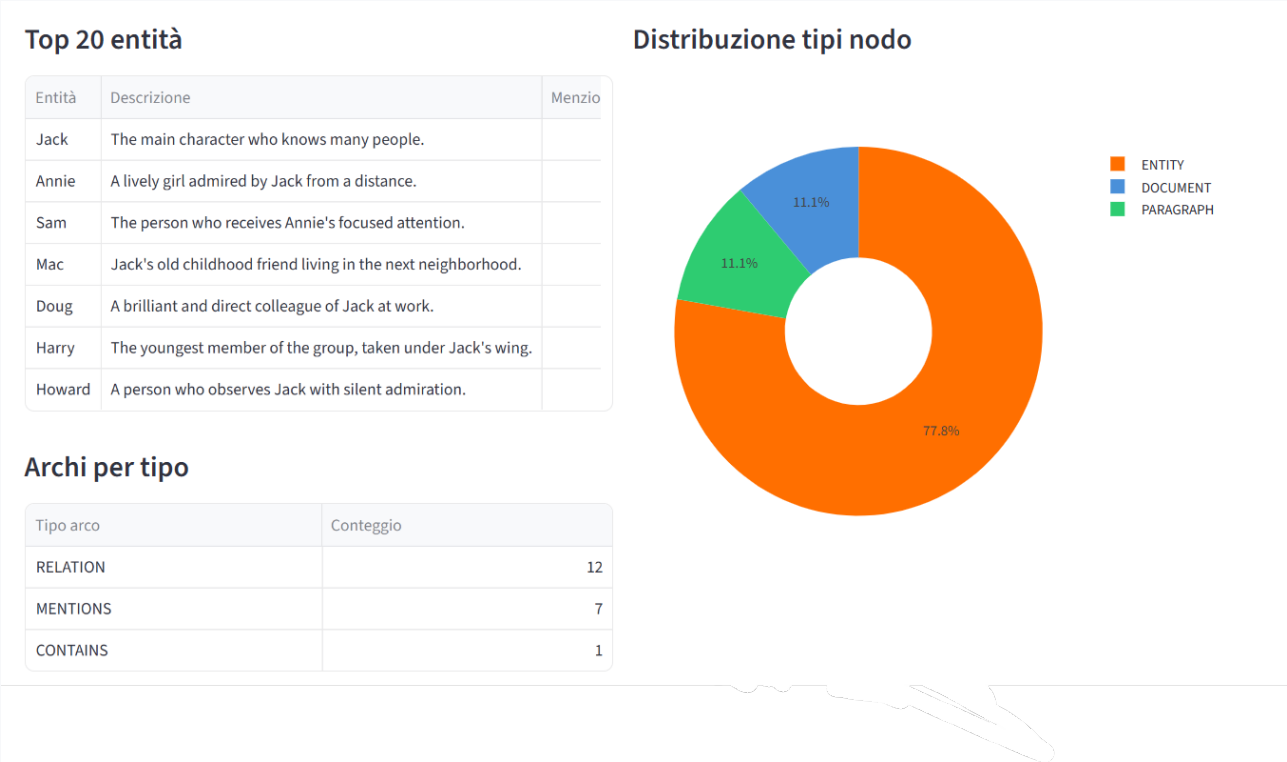

Figura 3: Dashboard dinamica per l'analisi e il monitoraggio delle metriche del grafo generato

Modelli utilizzati

Coerentemente con la filosofia dell'intero progetto, tutti i modelli impiegati vengono eseguiti in locale tramite OLLAMA [4], senza che alcun dato lasci l'infrastruttura dell'utente.

L'architettura è volutamente agnostica rispetto agli LLM impiegati: i modelli per l'estrazione di concetti ed entità, per l'analisi visiva delle immagini e per la disambiguazione dei duplicati sono configurabili tramite variabili d'ambiente, così come la dimensione della finestra di contesto e il numero di istanze parallele utilizzabili. Questo permette di adattare il sistema alle risorse hardware disponibili: configurazioni più potenti possono impiegare modelli di dimensioni maggiori e più worker in parallelo, mentre configurazioni più contenute possono ripiegare su modelli leggeri accettando tempi di elaborazione maggiori.

Nella configurazione di riferimento adottata per questo studio, ogni compito è affidato al modello più adatto al suo profilo di costo e complessità:

- **Gemma 4 E4B [5]:** dedicato alla costruzione dei grafi, impiegato per l'estrazione di concetti, entità e relazioni da testo e tabelle. Essendo il modello invocato con maggiore frequenza durante l'indicizzazione, la scelta è ricaduta su una variante compatta ed efficiente, in grado di sostenere l'elevato numero di chiamate necessarie.
- **Gemma 4 26B [6]:** modello multimodale utilizzato per l'analisi delle immagini, usato per generare le descrizioni e i concetti associati ai nodi IMAGE. Trattandosi del compito percettivamente più complesso, ma anche del meno frequente, è possibile impiegare un modello di dimensioni maggiori senza compromettere i tempi complessivi.
- **Llama 3.1 8B [7]:** riservato alla disambiguazione dei duplicati nella fase di collasso dei sotto-concetti, dove la richiesta si riduce a una risposta binaria e la velocità di inferenza è prioritaria rispetto alla capacità generativa.
- **mxbai-embed-large [8]:** modello di embedding (1024 dimensioni) utilizzato sia per l'indice vettoriale sia per il matching semantico tra la domanda dell'utente e i concetti del grafo, nonché per il riordinamento finale dei risultati.

Poiché i modelli di dimensioni contenute non sempre rispettano rigorosamente il formato di output richiesto, il sistema adotta infine una strategia di parsing tollerante: l'output JSON viene ripulito da eventuali delimitatori e ragionamenti espliciti, le virgole superflue vengono corrette e, nei casi più difficili, i singoli oggetti validi vengono recuperati anche da risposte parzialmente malformate. Ogni elemento estratto viene poi validato rispetto a una lista di esclusione di termini generici in italiano e inglese, che impedisce a parole come "documento", "sistema" o "data" di inquinare il grafo.

Sviluppi futuri

Lo studio conferma la validità dell'impiego degli LLM come strumento di strutturazione della conoscenza, estendendo l'approccio dalla normalizzazione dei documenti alla costruzione automatica di grafi di conoscenza, già esplorato in precedenza [3].

Un primo sviluppo naturale riguarda la convergenza tra il grafo concettuale e il grafo ER, oggi costruiti e mantenuti separatamente: una rappresentazione unificata permetterebbe al retrieval di sfruttare contemporaneamente la ricchezza relazionale del multigrafo e i meccanismi di attraversamento già maturi del grafo concettuale.

Un secondo filone riguarda la validazione quantitativa della modalità FAT RAG. Analogamente a quanto fatto per i moduli precedenti, si intende definire un protocollo di valutazione di questa nuova funzionalità. L'obiettivo è sia testare che confrontare sistematicamente il retrieval ibrido con quello puramente vettoriale su un insieme di interrogazioni a complessità crescente, misurando la qualità del contesto recuperato e l'impatto del parametro di fusione alpha sulle prestazioni complessive.

Ulteriori direzioni includono l'aggiornamento incrementale del grafo in caso di modifica o rimozione di documenti già indicizzati, l'impiego di modelli con finestre di contesto più ampie per ridurre la frammentazione delle sezioni lunghe, e l'estensione dei meccanismi di disambiguazione tramite LLM dall'attuale ambito dei sotto-concetti all'intero processo di riconciliazione delle entità.

Infine, l'introduzione di meccanismi di feedback umano (human-in-the-loop) sulla qualità dei concetti e delle relazioni estratte permetterebbe di affinare progressivamente i prompt di estrazione e le soglie di fusione, rendendo il sistema adattivo e migliorandone l'affidabilità nel tempo. Ciò rafforza la prospettiva, già delineata prima, degli LLM come componenti centrali per la costruzione automatica di basi di conoscenza strutturate a partire da fonti non strutturate.

Bibliografia

[1] Retrieval-Augmented-Generation (RAG) per la consultazione locale e sicura di documenti: https://t-defence.it/wp-content/uploads/2025/07/Report_AI4Cyber.pdf

[2] Retrieval-Augmented-Generation (RAG) per la consultazione locale e sicura di documenti: miglioramenti e nuove funzionalità: https://t-defence.it/wp-content/uploads/2025/08/Report_AI4Cyber.pdf

[3] Retrieval-Augmented-Generation (RAG) per la consultazione locale e sicura di documenti: una pipeline per la standardizzazione dei dati e lo scoring semantico: https://t-defence.it/wp-content/uploads/2026/01/260129-AI4Cyber-Retrieval-Augmented-Generation.pdf

[4] OLLAMA: https://ollama.com/

[5] Gemma 4 E4B: https://ollama.com/library/gemma4:e4b

[6] Gemma 4 26B: https://ollama.com/library/gemma4:26b

[7] Llama 3.1 8B: https://ollama.com/library/llama3.1:8b

[8] mxbai-embed-large: https://ollama.com/library/mxbai-embed-large

[9] GridFS: https://mongodb.github.io/node-mongodb-native/3.4/tutorials/gridfs/

[10] GraphML: https://graphml.ethz.ch/about.html

[11] Gephi: https://gephi.org/

[12] Cytoscape: https://cytoscape.org/

t-defence

Next | Donexit | Foramil | Innodesi

Via Giacomo Peroni, 452 - 00131 Roma
tel. 06.45752720 - info@defencetech.it
www.t-defence.it

#TDefenceBusiness