tinexta
defence

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

#TinextaDefenceBusiness

AI4Cyber

AI4Cyber studio, è il nuovo spazio di ricerca applicata di Tinexta 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**.

Summary

Abstract 04

1. Introduzione 05

2. Architettura del sistema 06

- 2.1 Modulo di indicizzazione 07
  - 2.1.1 Documenti strutturati 07
  - 2.1.2 Documenti non strutturati 08
  - 2.1.3 Documenti scansionati 08
- 2.2 Modulo di retrieval 08
  - 2.2.1 Modulo di ricerca per pagine 09
- 2.3 Modulo di generazione 10

3. Funzionalità avanzate del sistema 11

4. Validazione del sistema 12

5. Risultati 13

6. Limiti 14

7. Conclusioni e sviluppi futuri 15

8. Bibliografia 16

Abstract

Il presente studio illustra la realizzazione di un sistema che ha lo scopo di interrogare documenti aziendali attraverso l'utilizzo di tecniche di Intelligenza artificiale, in particolare di RAG (Retrieval Augmented Generation), sfruttando Large Language Model (LLM) in esecuzione locale, per offrire un assistente AI privato, in grado di garantire riservatezza, integrità e disponibilità dei dati sensibili.

La soluzione proposta integra un indice vettoriale e un corpus TF-IDF con l'obiettivo di costruire un contesto strutturato e accurato per il recupero delle informazioni.

La componente di Retrieval si basa sulla similarità semantica e sull'algoritmo TF-IDF, arricchiti da tecniche avanzate di pulizia e normalizzazione del testo, così da migliorare la precisione delle risposte e ridurre al minimo le ambiguità.

La generazione delle risposte, invece, è stata sviluppata tenendo conto degli esiti di una serie di test preliminari condotti su diversi modelli - tra cui LLama 3.1, Mistral e Phi - che hanno permesso di individuare in Phi il compromesso ideale tra efficienza e accuratezza.

Il sistema è stato validato su un insieme eterogeneo di documenti, ottenendo risposte piuttosto soddisfacenti. Tuttavia, sono emerse alcune eccezioni, in particolare in presenza di domande formulate in modo ambiguo o di richieste che implicano calcoli complessi, ambiti in cui l'output può risultare semplificato o impreciso.

A completare l'architettura contribuiscono un'interfaccia grafica intuitiva e la capacità di gestire documenti scansionati tramite OCR, elementi che rendono il sistema uno strumento flessibile e facilmente adattabile alle esigenze aziendali.

Autori:

- Simone Parrella: Analista AI4CYBER
- Simona Sorgente: Team Leader AI4CYBER

1. Introduzione

L'approccio RAG [10] sta emergendo come uno strumento fondamentale per la gestione e la consultazione di documenti, poiché consente di combinare la potenza degli LLM con la possibilità di estrarre informazioni da fonti testuali pertinenti. Tradizionalmente, i sistemi di recupero testuale si basavano su semplici algoritmi di ricerca o su tecniche di indicizzazione, spesso inadeguati per documenti complessi e strutturati in modo eterogeneo.

L'introduzione degli LLM ha aperto nuove possibilità nell'analisi e nella gestione documentale, grazie alla loro capacità di comprendere il linguaggio naturale e generare risposte contestuali. In questo contesto si inserisce il sistema proposto, che adotta un approccio innovativo basato su LLM on-premise, in grado di interrogare e sintetizzare contenuti provenienti da documenti strutturati, non strutturati o scansionati.

La scelta del tema di ricerca nasce dalla crescente esigenza, in ambito aziendale, di disporre di strumenti da usare in locale capaci di semplificare l'accesso a informazioni contenute in documenti voluminosi e complessi. Sebbene gli LLM offerti via cloud garantiscano buone prestazioni, il loro utilizzo solleva rilevanti problematiche legate alla privacy, in quanto richiedono l'invio di dati sensibili a server esterni.

Per rispondere a questa esigenza, si propone la realizzazione di un sistema completamente personalizzabile, che, pur necessitando di maggiori risorse hardware, consente non solo di mantenere il pieno controllo sui dati trattati ma anche di adattare il suo funzionamento rispetto a una vasta gamma di richieste.

2. Architettura del sistema

Il sistema sviluppato si basa su un'architettura modulare composta da tre componenti principali:

- Modulo di Indicizzazione: si occupa di analizzare e suddividere i documenti in base alla loro struttura. Nei documenti strutturati applica tecniche di analisi testuale per individuare paragrafi e contenuti; in quelli non strutturati estrae direttamente il testo digitale, mentre nei documenti scansionati utilizza il riconoscimento ottico dei caratteri (OCR) per convertire le immagini in testo. Il contenuto viene quindi suddiviso in porzioni chiamate chunk, memorizzate in un database vettoriale tramite embedding* e in un corpus TF-IDF locale*.

- Modulo di Recupero: riceve la query dell'utente e la rielabora per migliorarne la precisione. Combina similarità semantica e analisi TF-IDF per selezionare i chunk più rilevanti. Nel caso di documenti ben strutturati (ad esempio relazioni, report, manuali), recupera interi paragrafi per garantire un contesto più ampio. Il modulo si adatta anche a documenti con struttura debole o non formale (come appunti, trascrizioni o testi OCR), mantenendo comunque un buon livello di rilevanza nei risultati.

- Modulo di Generazione: è la componente che si occupa di generare le risposte. Dopo il recupero delle informazioni dal modulo precedente, genera una risposta mirata, basandosi esclusivamente sui contenuti dei documenti forniti. Nel caso specifico, il modello Phi-4 è stato selezionato per la sua capacità di mantenere un buon equilibrio tra accuratezza ed efficienza, garantendo risposte contestuali e coerenti.

**Embedding: è la rappresentazione numerica (vettore) di un testo che racchiude il significato delle parole o frasi, permettendo ai modelli di confrontare semanticamente testi diversi.*

**Corpus TF-IDF: è una raccolta di documenti trasformata in una matrice basata su quanto spesso appaiono le parole (Term Frequency=TF) e quanto sono rare nei documenti (Inverse Document Frequency= IDF), utile per identificare termini rilevanti in un testo.*

2.1 Modulo di indicizzazione

La fase di indicizzazione ha lo scopo di rendere i documenti rapidamente ricercabili, trasformandone il contenuto in una rappresentazione numerica. Il processo inizia con l'estrazione del testo dai file, un passaggio essenziale per il corretto funzionamento delle fasi successive. I documenti possono presentarsi in forma strutturata o non strutturata, tuttavia, quando i documenti sono acquisiti tramite scansione e processati da un sistema OCR, la struttura originaria può andare persa o risultare alterata, rendendo meno chiara questa distinzione per il sistema.

Se il documento è strutturato, viene suddiviso in paragrafi che vengono poi frammentati in porzioni di lunghezza controllata (chunk) tramite RecursiveCharacterTextSplitter di LangChain [6], così da garantire coerenza negli embedding generati. Per i documenti non strutturati o scansionati, dove la suddivisione in paragrafi può essere assente o compromessa, si utilizza una funzione alternativa che segmenta il testo in chunk sulla base di delimitatori predefiniti ("\n\n", "\n", ".").

Gli embedding risultanti vengono inseriti in una struttura vettoriale tramite FAISS [5] che consente ricerche basate sulla similarità semantica. Infine, per ogni chunk indicizzato, vengono salvate informazioni aggiuntive come il nome del file, la sezione di appartenenza (nel caso di documenti strutturati), il tipo di contenuto e il testo originale.

2.1.1 Documenti strutturati

Nel caso di documenti strutturati, come quelli composti da paragrafi ben definiti, si implementa una strategia di analisi basata sulla dimensione del font più ricorrente (moda). Questo consente di identificare le principali sezioni logiche del testo (ad esempio capitoli, titoli e paragrafi), separandole in modo coerente. Ogni sezione così individuata viene associata alle pagine da cui proviene ed etichettata come strutturata.

Questa tecnica si rivela particolarmente efficace nel trattamento di PDF nati digitalmente e con una gerarchia visiva stabile, in cui gli elementi come i titoli si distinguono chiaramente dal corpo del testo, ad esempio tramite l'uso di un font più grande.

2.1.2 Documenti non strutturati

Quando il documento contiene testo, ma non presenta una formattazione coerente, ad esempio, PDF generati da scansioni che però consentono l'estrazione del testo o documenti digitali generati in modo non convenzionale (come esportazioni da strumenti che non mantengono una struttura definita), esso viene trattato come un unico blocco testuale. Il contenuto viene quindi sottoposto a un processo di pulizia, che utilizza espressioni regolari per individuare e rimuovere caratteri non significativi, elementi di disturbo o "rumore" (come spazi bianchi e simboli anomali). Successivamente, il documento viene classificato come non strutturato in base all'assenza di una gerarchia o formattazione coerente.

2.1.3 Documenti scansionati

Nel caso in cui il documento non sia digitalizzato, il contenuto testuale viene estratto tramite OCR Tesseract [2], quindi sottoposto a un processo di pulizia e trattato come previsto per i documenti non strutturati.

Inoltre, per evitare errori di lettura dovuti a scansioni poco leggibili, è stata impiegata la tecnica di binarizzazione di Otsu [3] che consente di distinguere chiaramente il testo dal resto.

2.2 Modulo di retrieval

Il modulo di retrieval combina tecniche basate su embedding e TF-IDF [4]. Una volta ricevuta la query, questa viene duplicata ed espansa, poi da entrambe le versioni (originale ed espansa) si creano ulteriori due copie rimuovendo le stopword. Si ottengono così quattro varianti della query, utilizzate per l'estrazione dei chunk più rilevanti tramite embedding. Successivamente, su queste quattro varianti viene applicato anche l'algoritmo TF-IDF, al fine di ottenere ulteriori risultati che possano confermare o migliorare il recupero iniziale basato sugli embedding.

Questo approccio varia a seconda del tipo di documento. Nei documenti strutturati, infatti, non si utilizzano direttamente i singoli chunk, ma si risale al paragrafo completo da cui essi provengono, identificandolo tramite il titolo associato, così da ottenere un contesto più preciso. Nei documenti non strutturati, invece, il sistema estrae direttamente i chunk ritenuti più rilevanti e li restituisce come contesto.

2.2.1 Modulo di ricerca per pagine

Oltre al tradizionale approccio domanda-risposta basato sull'intero contenuto indicizzato, il sistema è in grado di gestire richieste riferite a pagine specifiche o a intervalli di pagine. Questo consente all'utente di ottenere risposte circoscritte esclusivamente al contenuto di determinate sezioni del documento, migliorando la precisione e la pertinenza dell'informazione restituita.

Per implementare questa funzionalità, è stato sviluppato un modulo che analizza la query dell'utente tramite espressioni regolari, progettate per riconoscere una varietà di pattern linguistici comunemente utilizzati per indicare riferimenti a pagine (es. "pagina 5", "dalla 3 alla 7", "pagine 10, 12 e 14"). Una volta identificati questi riferimenti, il sistema li normalizza in un formato interno standard (una lista di numeri di pagina) che viene poi utilizzato per filtrare e selezionare solo i paragrafi o i chunk corrispondenti a quelle pagine all'interno della base di conoscenza.

2.3 Modulo di generazione

Il modulo di generazione è responsabile della produzione della risposta finale a partire dal contesto testuale fornito dal modulo di recupero.

Questo componente si basa su un LLM eseguito localmente tramite l'API di Ollama [1].

Dal punto di vista architetturale, il modulo costruisce dinamicamente un prompt strutturato, composto da due parti principali:

- Il System Prompt che definisce le istruzioni operative da seguire per la generazione della risposta (ad es. adottare uno stile diretto e conciso, evitare inferenze non supportate dal contesto, attenersi esclusivamente alle informazioni fornite) e il contesto;

- L' User Prompt che contiene la query dell'utente.

Il prompt completo viene quindi inviato tramite una richiesta POST all'interfaccia locale di Ollama, che elabora l'input e genera una risposta.

3. Funzionalità avanzate del sistema

Il sistema implementato è in grado di gestire:

- PDF strutturati: tramite parsing del testo e identificazione delle sezioni basata sulla moda della dimensione del font.
- PDF non strutturati: se non viene rilevata una struttura coerente nei font, viene attivata automaticamente una modalità alternativa di estrazione del contenuto.
- PDF scansionati: attraverso un sistema di riconoscimento OCR.
- Ricerche su pagine specifiche: se vengono richieste una o più pagine, viene utilizzata una funzione dedicata per isolare la ricerca sui soli intervalli indicati.
- Ogni chunk contiene metadati utili come:
  - Titolo della sezione.
  - Pagine di riferimento.
  - Indicazione della struttura (structured: yes/no), che può guidare strategie diverse nel retrieval.
- Per garantire una comprensione ottimale da parte del modello, viene eseguita una pulizia intelligente del testo, con rimozione di contenuti inutili (come INDICI, SOMMARI o intestazioni ripetute) e normalizzazione linguistica avanzata (Unicode, punteggiatura, spaziature, etc).

4. Validazione del sistema

I modelli impiegati nella sperimentazione sono stati:

- LLaMA 3.1 [7]
- Mistral [8]
- Phi-4 [9]

Modello	Parametri	Dimensione (GB)
LLama 3.1	8B	4.9GB
Mistral	7B	4.1GB
Phi-4	14B	9.1GB

La prototipazione è stata effettuata utilizzando PHI-4, scelto a seguito di una serie di esperimenti eseguiti su un sottoinsieme di documenti: i test, condotti sui tre modelli, hanno esplorato diverse combinazioni di Chunk Size*, Chunk Overlap* e numero di chunk restituiti* per ciascun set di query (composto dalla query originale e dalle tre riformulazioni derivate).

CHUNK SIZE	CHUNK OVERLAP	CHUNK RESTITUITI
50	5	1
100	15	5
256	30	10
512	70	15

**Chunk Size: È la dimensione (in numero di caratteri o parole) di ciascun blocco di testo in cui un documento viene suddiviso per l'elaborazione.*

**Chunk Overlap: Indica quanti caratteri o parole vengono ripetuti tra un chunk e il successivo.*

**Numero di chunk restituiti: È il numero massimo di chunk selezionati dal sistema come rilevanti per una query.*

5. Risultati

Per valutare le prestazioni del sistema è stato utilizzato il Bert score e un analista umano.

Documento	Accuratezza	N° di domande	Tipo documento
Documento aziendale 1	86.53%	26	STRUTTURATO
Documento aziendale 2	100%	16	STRUTTURATO
Documento aziendale 3	75.8%	29	STRUTTURATO
Documento aziendale 4	100%	12	STRUTTURATO
Documento a tema informatica	100%	15	STRUTTURATO
Documento a tema economico	100%	13	NON STRUTTURATO
Documento storico	95%	20	NON STRUTTURATO
Documento storico	100%	15	NON STRUTTURATO
Documento storico	92.3%	13	NON STRUTTURATO
Documento aziendale 5	100%	15	NON STRUTTURATO
Documento aziendale 6	100%	12	SCANSIONATO
Documento di statistica	92.3%	13	SCANSIONATO
Documento scientifico	85%	20	SCANSIONATO
Documento culturale	93.3%	15	SCANSIONATO
Documento storico	100%	12	SCANSIONATO

Complessivamente, i risultati ottenuti sono ottimi. Solo in pochi casi sono state rilevate risposte errate, dovute a richieste di calcoli complessi o a formulazioni ambigue, che hanno portato il modello a produrre output semplificati o imprecisi.

6. Limiti

Durante la fase di testing sono emerse alcune criticità tipiche dei sistemi basati su RAG.

- Prestazioni hardware e modelli: Il RAG si conferma una metodologia estremamente potente e versatile ma l'accuratezza dei risultati è strettamente legata alla qualità dell'hardware e del modello utilizzato. Modelli ad alta precisione richiedono risorse computazionali elevate; in alternativa, è possibile utilizzare modelli più leggeri con hardware meno performante, a scapito però della precisione e della complessità delle richieste gestibili.

- Struttura dei documenti: L'organizzazione dei contenuti nei documenti ha un impatto determinante sul funzionamento del sistema. Una struttura chiara e coerente facilita l'estrazione del contesto e semplifica il processo di retrieval, riducendo il rischio di perdere informazioni rilevanti.

- Chiarezza delle domande: Il sistema non interpreta intenzioni implicite ma opera esclusivamente sulla base della semantica delle domande e del contesto disponibile. È quindi fondamentale che le domande siano formulate in modo chiaro, diretto e preciso.

- Ridondanza documentale: Un'eccessiva presenza di documenti con contenuti simili può generare ambiguità, compromettendo l'affidabilità delle risposte, anche quando i documenti recuperati sono in linea con la richiesta.

- Documenti scansionati: La qualità visiva dei documenti scansionati incide fortemente sull'efficacia dell'OCR. Se un contenuto risulta poco leggibile per l'essere umano, è probabile che lo sia anche per un sistema di riconoscimento ottico.

- Limitazioni dei modelli generativi: I modelli di generazione testuale di fascia bassa non sono in grado di risolvere calcoli complessi. Per questo motivo, le risposte a quesiti matematici devono essere sempre verificate. Inoltre, la lingua italiana rappresenta ancora una sfida per molti modelli di embedding, che sono spesso ottimizzati principalmente per l'inglese.

7. Conclusioni e sviluppi futuri

Il sistema sviluppato si è rivelato una soluzione efficace per velocizzare la ricerca e l'estrazione di informazioni da documenti aziendali, contribuendo a migliorare l'efficienza delle attività quotidiane.

Sebbene i risultati ottenuti siano incoraggianti, esistono ancora margini di miglioramento, in particolare nella gestione di documenti non strutturati o scansionati, dove la qualità dell'estrazione può risentire della complessità del formato.

Nel complesso, questa tecnologia rappresenta un valido strumento per ottimizzare il lavoro e facilitare l'accesso rapido alle informazioni. Tuttavia, l'aspetto più rilevante dell'implementazione di un sistema locale di questo tipo è la possibilità di disporre di un assistente AI privato, capace di supportare una vasta gamma di attività rispettando i principi fondamentali della triade CIA: Confidenzialità, Integrità e Disponibilità.

Considerando l'attuale evoluzione dell'intelligenza artificiale, diventa sempre più cruciale garantire la protezione dei dati personali e sensibili. I sistemi basati su cloud, infatti, possono esporre i dati a rischi di intercettazione o manipolazione durante la trasmissione.

Nel contesto in cui si cala il sistema proposto, l'utilizzo di modelli locali, sebbene spesso meno performanti rispetto a soluzioni come GPT, rappresenta una scelta più sicura, poiché consente un maggiore controllo sui dati e sul canale di trasmissione, risultando particolarmente adatta in scenari in cui sono richiesti elevati livelli di riservatezza.

8. Bibliografia

1. Ollama: https://ollama.com/

2. Tesseract: https://github.com/tesseract-ocr/tesseract

3. Binarizzazione di OTSU: https://arxiv.org/abs/2007.07350

4. TF-IDF: Introduction to Information Retrieval.
Cambridge University Press, 2008.
ISBN: 978-0521865715.

5. FAISS: https://python.langchain.com/docs/integrations/vectorstores/faiss/

6. LangChain: https://python.langchain.com/docs/get_started/introductionMeta

7. Llama 3: https://ollama.com/library/llama3.1

8. Mistral AI: https://ollama.com/library/mistralPhi4

9. Microsoft Phi-4: https://ollama.com/library/phi4

10. Yunfan Gao et al. “Retrieval-augmented generation for large language models: A survey”. In: arXiv preprint arXiv:2312.10997 (2023).

tinexta
defence

Defence Tech | Next |
Donexit | Foramil | Innodesi

Via Giacomo Peroni, 452 – 00131 Roma
tel. 06.45752720 – info@defencetech.it
www.tinextadefence.it

#TinextaDefenceBusiness