Strategie per la preparazione dei documenti in Generative AI

Ing. Raffaele Rialdi
Freelancer
&
CTO @ Vevy Europe

>>AI CONF 2025

improove

Kudos++

Diamond

Gold

Bronze

SPONSOR

adesso.it

FairMind

KEY2

ALPITOURWORLD

Red Hat

PARTNER

AI

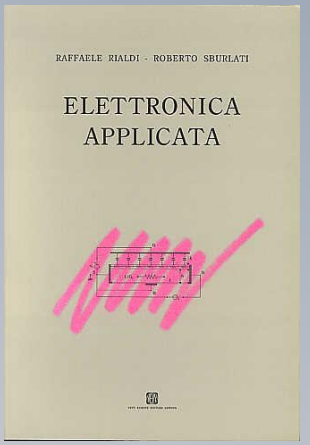
Cloud Champions

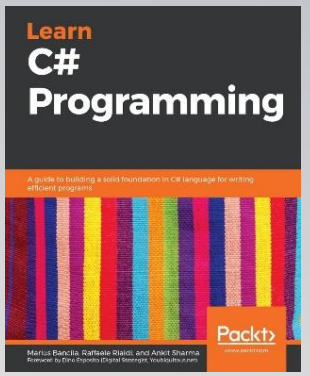
Google Developer Groups Genova

UGIdotNET

Chi sono

- Laurea Master in Ingegneria Elettronica (Unige)
- Insegno saltuariamente a Ingegneria Informatica (Unige)
- Membro della commissione ICT dell'Ordine degli Ingegneri
- Libero professionista, Software Architect Consultant in diverse aree:
  - Financial, Manufacturing, Healthcare, F1 racing, ...
- Speaker in conferenze nel mondo (più di 200 interventi in 20 anni)
  - Europa, Asia, USA
- Co-Autore di "Elettronica Applicata" e "C# Programming"
- Presidente di DotNetLiguria, community attiva a Genova e in Liguria
- Microsoft Most Valuable Professional per 21 anni consecutivi



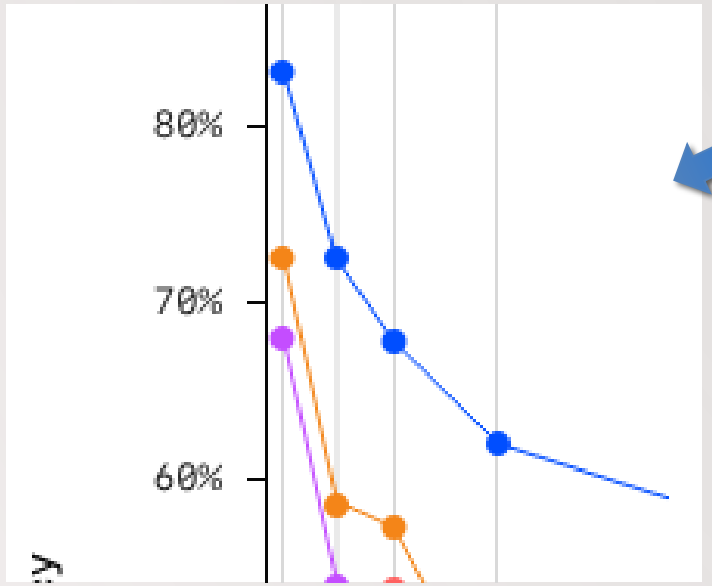


Ingestione dei documenti: quali requisiti?

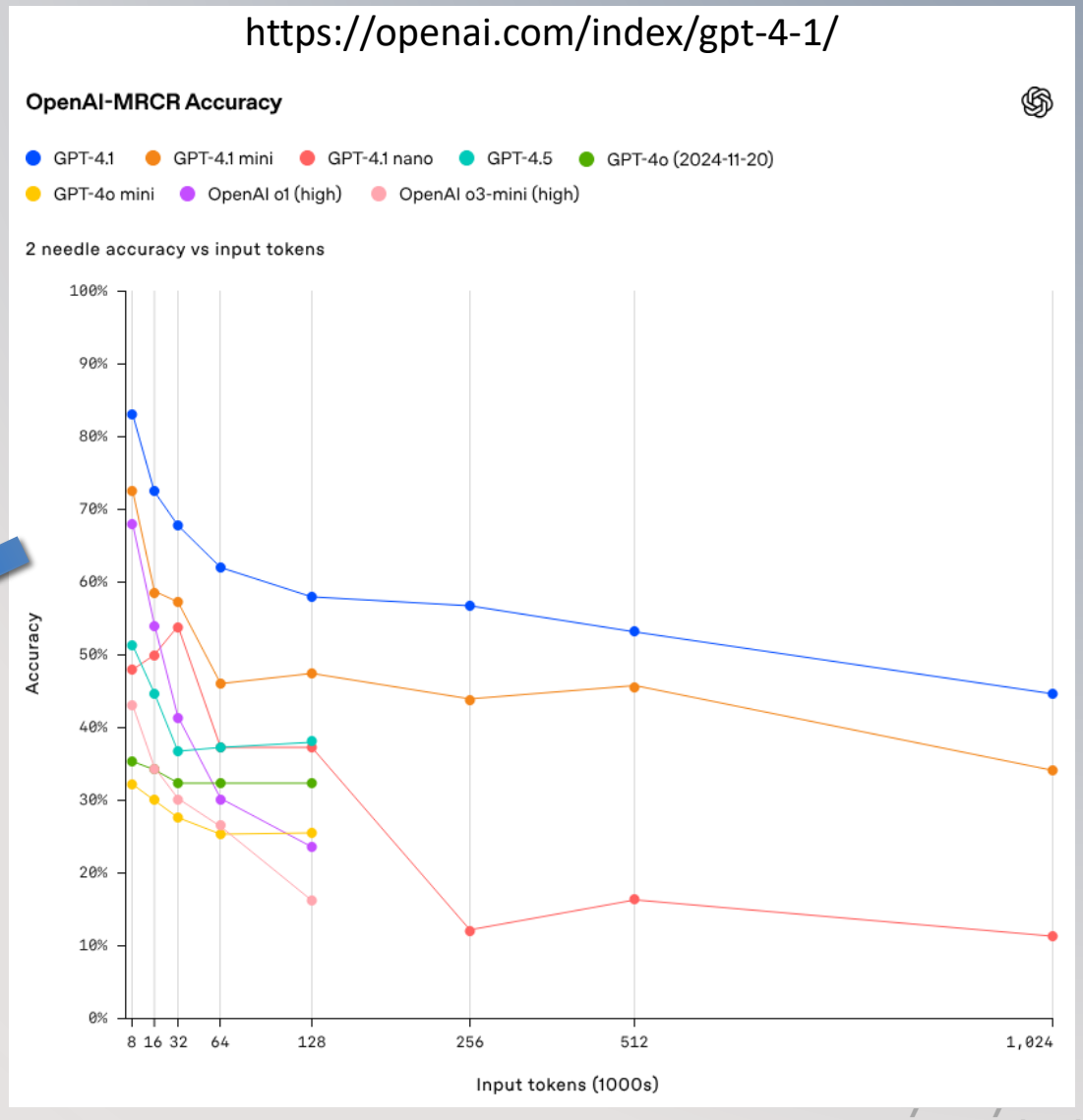
- La preparazione dei documenti cambia a seconda di:
  - Tipo di documento (pdf, docx, xlsx, pptx, txt, database, …)
  - Prompt dell'utente
  - Scenario / use-case (medico, industriale, scientifico, legale, …)
  - Il ciclo di vita dei documenti (variazioni nel tempo)
- Quale parte del sistema è impattato?
  - Tecnica utilizzata (RAG, CAG, Knowledge Tree, Index-free, …)
  - Trasformazioni ai documenti. **Il goal è ottenere dei cluster di argomenti**
  - Modello di Embedding da utilizzare
  - Modifica al prompt utente per contestualizzare meglio
- Difficilmente un sistema generico ottiene risultati migliori del verticale.

RAG senza Embeddings?

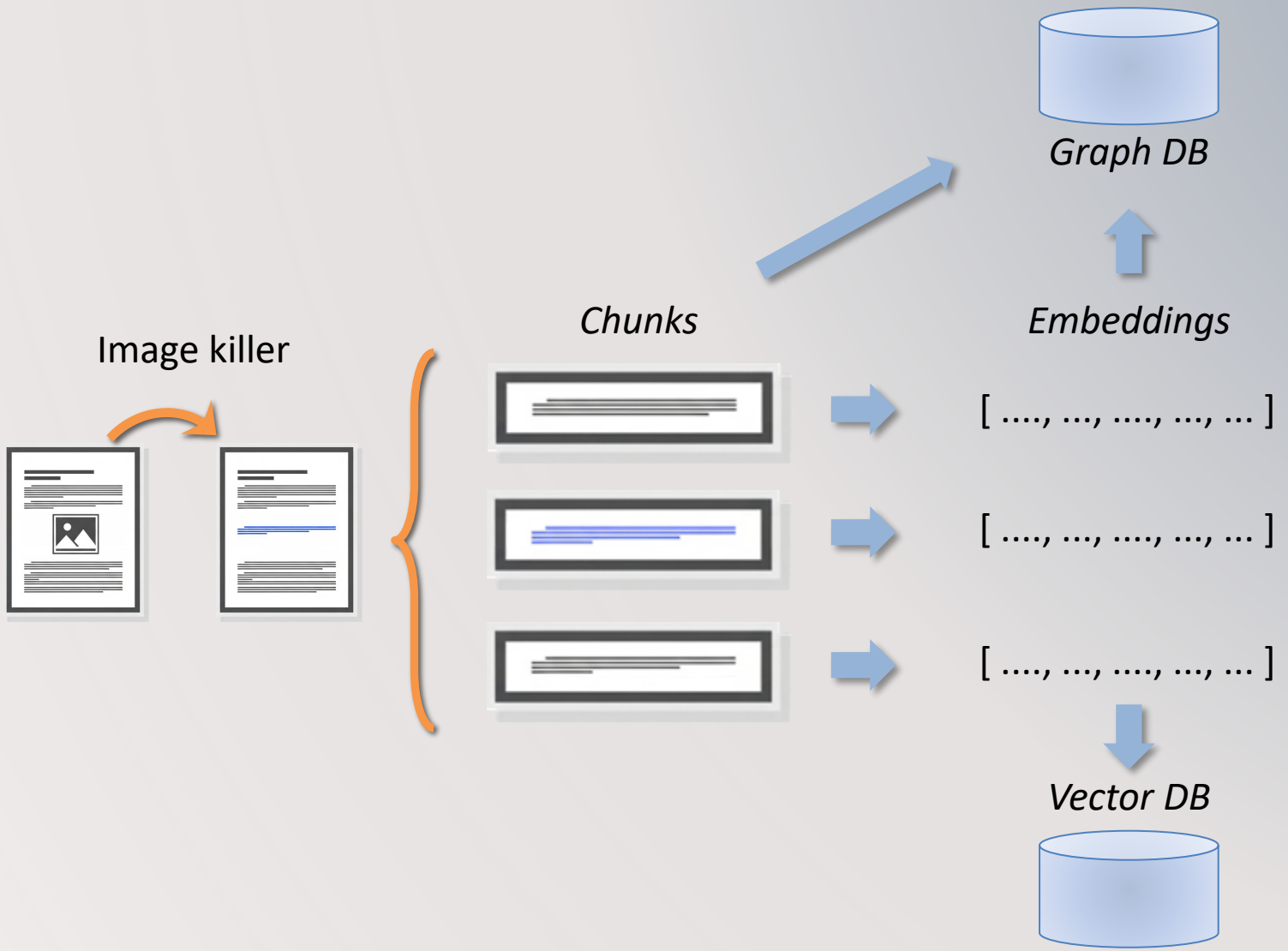
- OpenAI parla di Index-free RAG
- GPT 4.1 arriva ad un contesto di 1M token
- Sopra i 128K è molto stabile
- Da 8 a 64K la **perdita** di accuratezza è davvero notevole.



https://openai.com/index/gpt-4-1/



Preparazione del documento



Convertire i documenti in Markdown

Da PDF a Markdown

- I file PDF sono pensati per la stampa. Si usano solo come ultima risorsa
  - Lo stesso documento può essere scritto in molti modi differenti
  - Le font possono essere state offuscate
- Le librerie più note sono:
  - **https://github.com/docling-project/docling (IBM) [Multi-formato]**
  - https://github.com/microsoft/markitdown (Microsoft) [Multi-formato]
  - https://github.com/pdfminer/pdfminer.six
  - https://github.com/Belval/pdf2image
  - https://github.com/py-pdf/pypdf
  - https://github.com/datalab-to/marker
  - https://github.com/cuuupid/cog-marker
  - Pandoc, PyMuPDF4LLM, pdf2markdown4llm

Librerie per la conversione custom di documenti Office

- Lettura metadati del documento (autore, etc.): [docx, openpyx, pptx]
- Microsoft Word
  - [mammoth] Conversione da docx a html
  - [markdownify] Conversione da html a markdown
  - Rispettare la gerarchia dei titoli e preservare i paragrafi nel modo corretto
- Microsoft Excel
  - [openpyxl] Lettura isole di dati in Excel
  - [custom] Creazione elementi Markdown (tabelle)
    - Due passaggi per formule e valori calcolati (richiede ricalcolo formule / COM interop)
  - Opzione 1: creare un fraseggio per ogni tabella nello spreadsheet
  - Opzione 2: creare relazioni basate sulle formule.

Microsoft Powerpoint

- Estrarre testo e shape è facile con 'pptx'
  - Conservarne il **significato semantico** non lo è
- Soluzione:
  - Processare l'intera slide come immagine
  - Gli LLM comprendono anche il testo
  - Ottimo per rendere **accessibile** il documento!
- LLama3.2-vision fa un ottimo lavoro
- Per creare le immagini è necessario:
  - Windows e Powerpoint
  - Interoperabilità COM (Python: win32com.client)
  - Metadati (Python: pptx).

Ricordate questa slide?

- ImageKiller al lavoro ...
  (La soluzione all'accessibilità)

{ "file": "slide_13.png",
"hash": "a66ab93b9dc8d3c2eca0d0f84192b199251efa10f29da0bd88bc40f610bf2fd4",
"model_llama3.2-vision": "This infographic, titled \"Preparazione del Documento\" in blue text at the top left, provides an overview of data preparation processes for a document. The flowchart is organized from left to right, with arrows indicating the progression of steps.\n\nOn the far left, two images are shown: one labeled \"Image Killer,\" likely referencing image removal or extraction software, and another featuring three documents. An orange arrow connects these images to an arrangement of four boxes on the right side, labeled \"Chunks.\" These chunks are connected by blue arrows, which lead to a box labeled \"Graph DB\" and two others, \"Vector DB\" and \"Embeddings,\" all located in the center-right.\n\nThe background features a gradient that transitions from white at the left edge to dark grey at the right. A small grey watermark reading \"@raffaeler\" is situated near the bottom right corner of the image. Overall, this infographic provides a clear visual representation of the data preparation process for a document, highlighting the key steps involved in preparing the data for use."}

Processare i Markdown per l'ingestione

Immagini nei documenti

- Due diverse strategie:
  - Uso di modelli di Embedding multimodali
  - Sostituzione dell'immagine con una descrizione generata da un LLM
- LLama3.2-vision è un buon modello
  - Problema: inizia sempre la frase con: "Questa immagine rappresenta" ...
  - Optional: riscrivere la descrizione con un secondo modello
- I tag image in markdown sono 2
  - ![reference](file/url/base64)
  -

Strategie di Chunking del documento

- Le strategie non-semantiche sono perdenti per definizione
  - Fixed-size, sliding-window con overlapping
- Il chunking semantico è pregevole ma ha un grosso difetto
  - Se il contesto è composto da più elementi semantici, si rischia di perderlo
    - Esempio: Il cappuccino questa mattina era molto buono. Molto dipende dal barista che questa mattina ha dato davvero il massimo. L'aroma era davvero di prima qualità.
- Una strategia diversa (custom):
  - Il paragrafo (se ben scritto) aggrega i concetti che l'autore voleva esprimere
  - Il titolo (o la gerarchia dei titoli) ha una forte valenza per il contesto di ogni paragrafo

Comprendere il file Markdown (scritto bene)

- La libreria *mistletoe* produce il **syntax tree** modificabile del Markdown
  - È la libreria usata per rimpiazzare le immagini con la loro descrizione
- Una prima suddivisione in paragrafi, liste e tabelle
  - Ogni "blocco" accumula gli altri elementi "figli"
- Opzionale: **iniettare in ogni blocco lo stack dei titoli corrente**
- Conteggio del numero di token di ogni blocco
  - Se la dimensione supera una soglia, estrazione delle **frasi** tramite libreria *NLTK*
  - Redistribuzione delle frasi:
    - Numero minore possibile di blocchi
    - Numero di token omogeneo per ciascun blocco.

I modelli di Embedding

MTEB Leaderboard (al 22 Giugno)

M = MRL: Matryoshka Representation Learning support (dimensione custom)

Rank (Bo…	Model		Zero-shot	Memory U…	Number of P…	Embedding D…	Max Tokens
1	gemini-embedding-001	M	99%	Unknown	Unknown	3072	2048
2	Qwen3-Embedding-8B	M	99%	28866	7B	4096	32768
3	Qwen3-Embedding-4B	M	99%	15341	4B	2560	32768
4	Qwen3-Embedding-0.6B	M	99%	2272	595M	1024	32768
5	Linq-Embed-Mistral	X	99%	13563	7B	4096	32768
6	gte-Qwen2-7B-instruct	X	⚠ NA	29040	7B	3584	32768
7	multilingual-e5-large-instruct	X	99%	1068	560M	1024	514
8	SFR-Embedding-Mistral	?	96%	13563	7B	4096	32768
16	text-embedding-3-large	M	⚠ NA	Unknown	Unknown	3072	8191

https://huggingface.co/spaces/mteb/leaderboard

Modelli di Embedding di tipo Instruct

- I modelli di Embedding di tipo Instruct usano il "**dual encoder**"
  - Durante il training domande e risposte sono immessi in due decoder differenti
  - Il training premia la risposta positiva dove "Instruct" e "Query" sono parole speciali
- La "Query1" funziona meglio se l'utente fa una domanda
  - Il loro mancato uso può comportare una perdita tra 1% e 5%
- La tecnica del "Prompt Rewriting" è utile per generare il prompt corretto

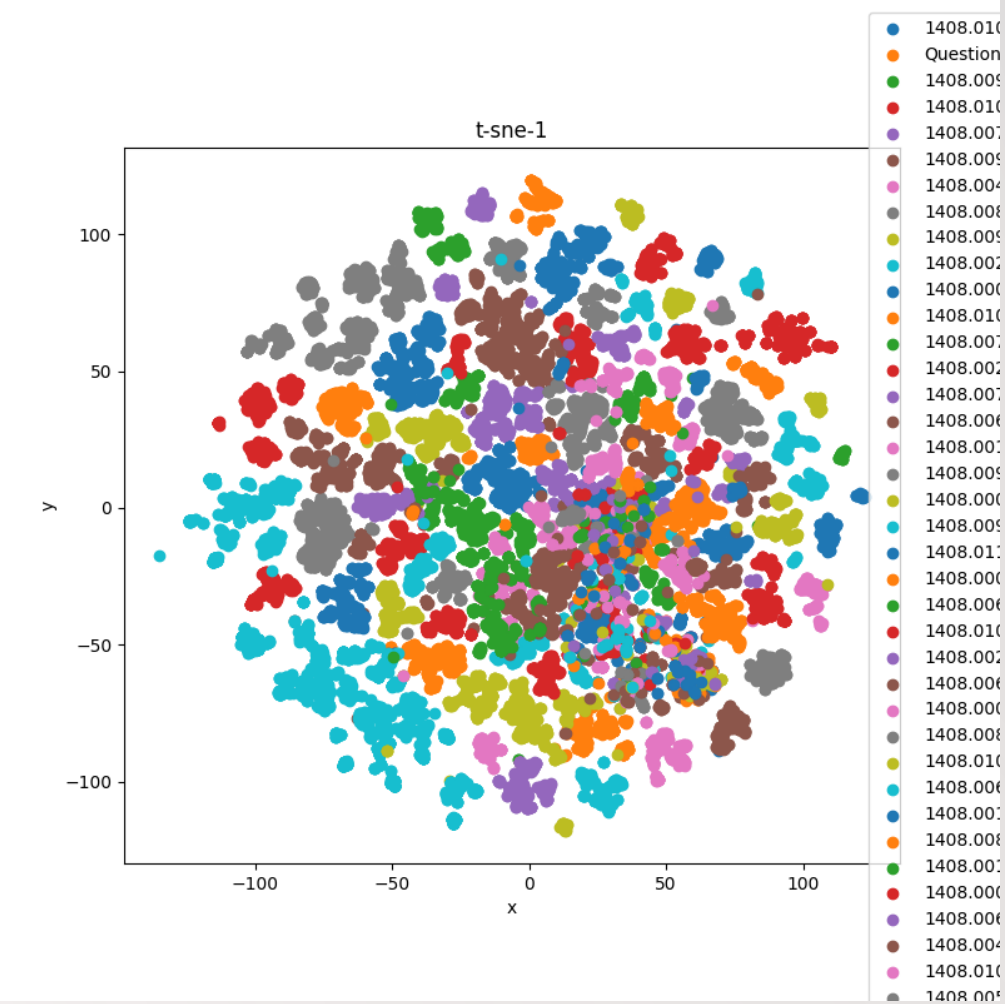
Documento	The capital of China is Beijing.	Qwen 0.6B	Qwen 8B
Query 1	Instruct: Given a question, retrieve the relevant sentences answering the question. Query: What is the capital of China?	0.8457031	0.7832031
Query 2	What is the capital of China?	0.8022461	0.8842773

Calcolo degli Embeddings con la libreria Transformer

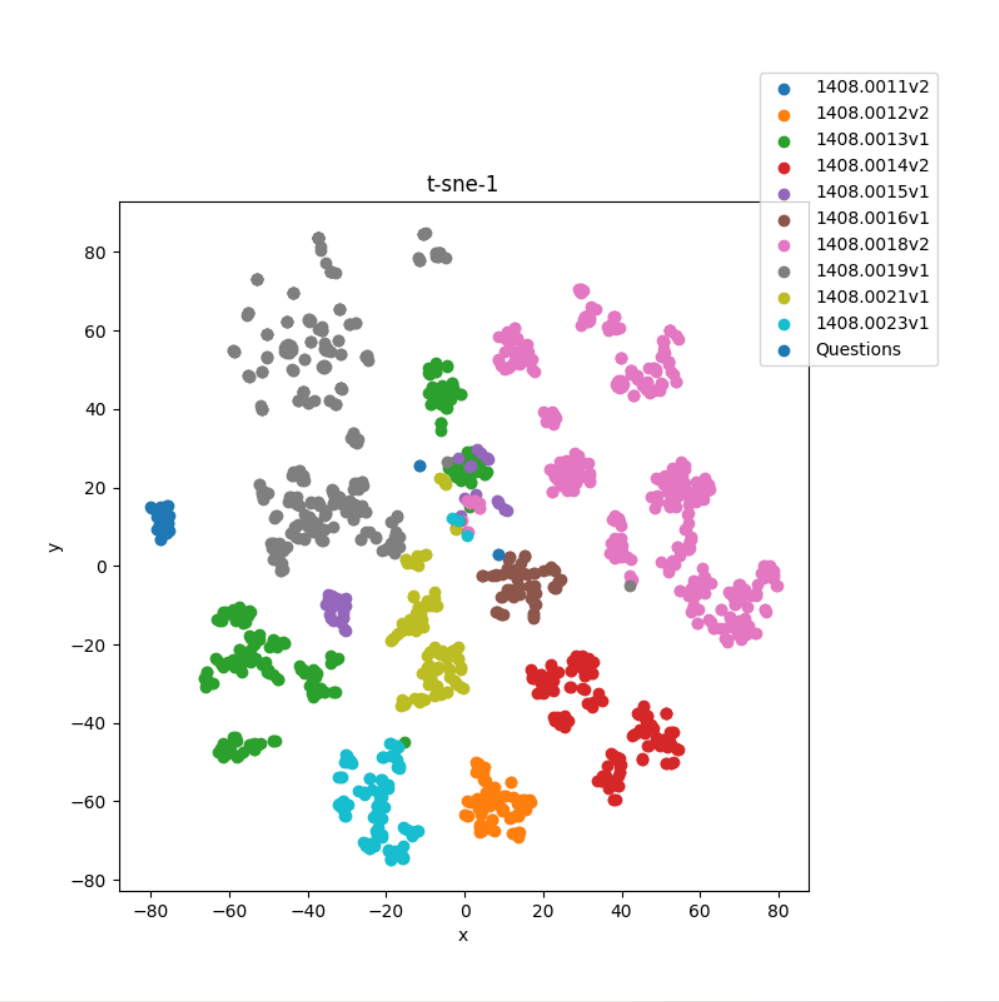
- Usando la libreria "Transformer", resettare l'Attention ad ogni chunk
  - Diversamente il calcolo in batch degli Embedding è influenzato dai precedenti!
- A seconda del modello e dell'uso, si usa una tecnica di Pooling diversa
- Pooling: metodo per estrarre l'Embedding dall'ultimo layer della rete
  - "Last Token Pooling": utilizza il contesto dell'ultimo token
    - Presuppone che la rete abbia "propagato" il contesto all'ultimo token
  - "Average": esegue la media dei token sull'ultimo layer
  - "Max": seleziona il valore massimo ignorando il contesto di valore inferiore
  - "Token [CLS]": seleziona il valore del toke [CLS] all'inizio (modelli basati su BERT)
  - "Attention": media dei pesi di ciascun token in uscita dall'Attention.

Risultati

100 Articoli presi da Arxiv

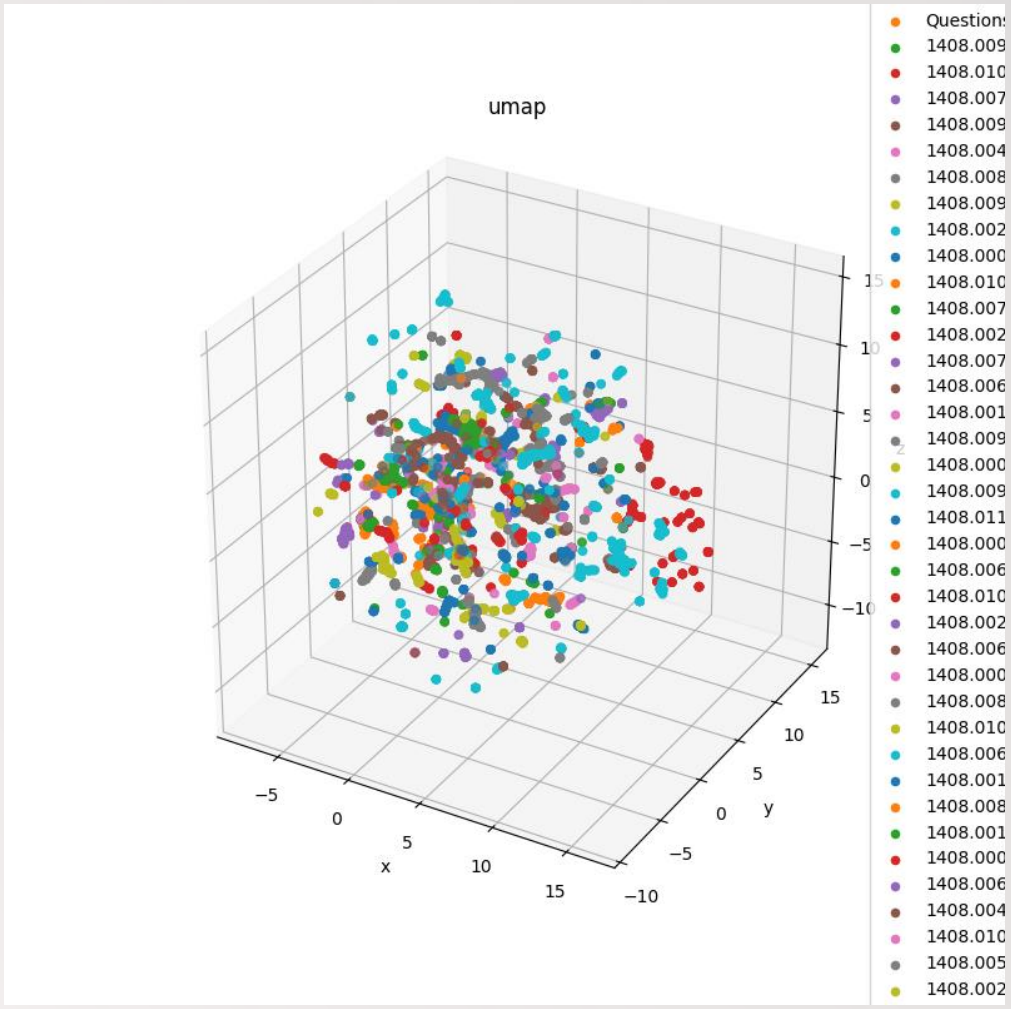


10 Articoli presi da Arxiv

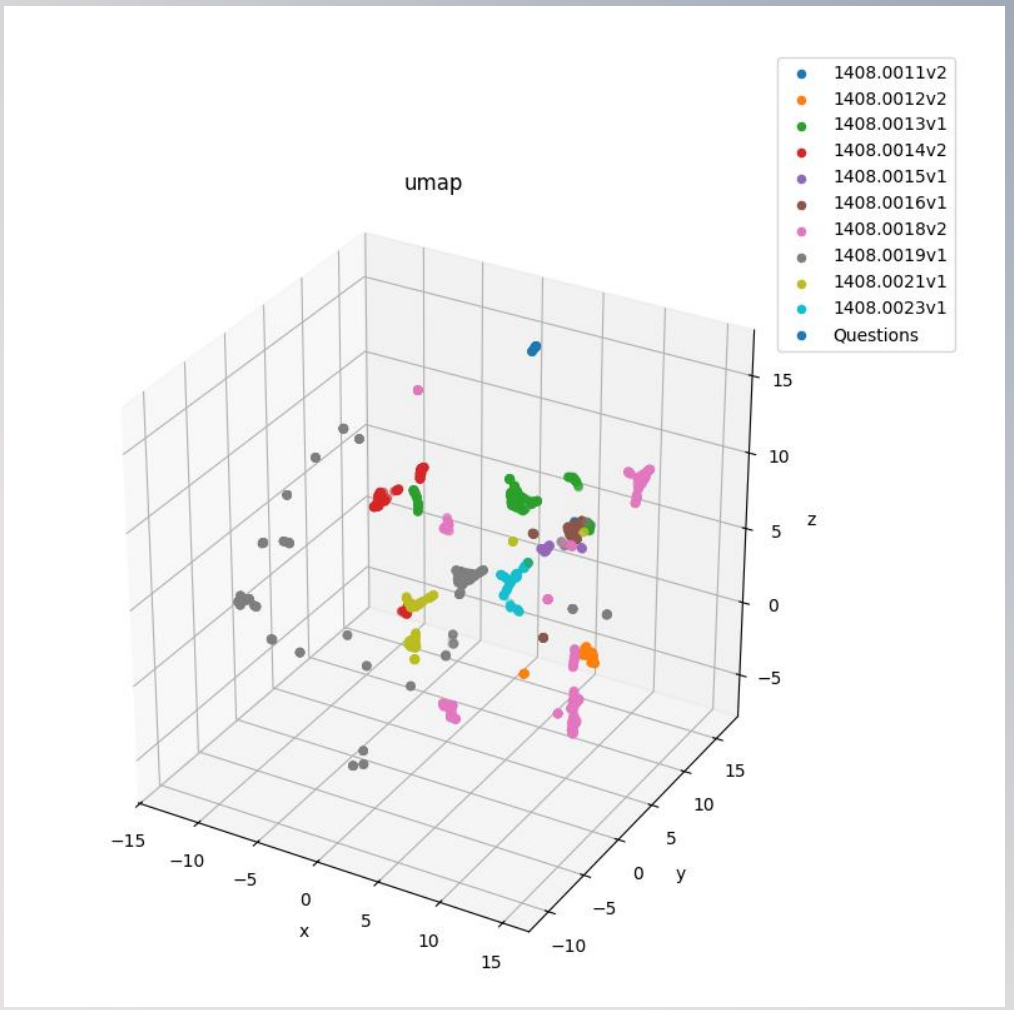


Risultati

100 Articoli presi da Arxiv



10 Articoli presi da Arxiv



Conclusioni

- Conversione dei documenti di qualità
- Scegliere il giusto modello di Embedding
- Migliorare i risultati con i modelli di Re-Ranking
- Visualizzate i vostri cluster
- Il futuro va verso i modelli Instruct
- Ipotizzo che futuri modelli di Embedding usino MoE
  - MoE = Mixture Of Experts.

Thank you!

👉 slides & videos: https://www.improove.tech/videos

>>AI CONF 2025

improove