

UNIVERSITÀ DEGLI STUDI DI SALERNO

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE ED
ELETTRICA E MATEMATICA APPLICATA

Corso di Laurea Magistrale in Ingegneria Informatica



FINAL PROJECT 2024/2025

NATURAL LANGUAGE PROCESSING E LARGE LANGUAGE MODELS

GRUPPO 7

DOCENTI

Nicola CAPUANO

Antonio GRECO

STUDENTI

Valentina MICERA – 0622702379

Raffaele SBARDELLA – 0622702312

Jacopo VOLPE – 0622702301

Egidio ZOTTARELLI – 0622702175

ANNO ACCADEMICO 2024/2025

Sommario

1.	Introduzione	3
2.	Architettura del Sistema	4
3.	Preparazione dei Dati	7
	Fonti dei Dati	7
	Fasi del pre-processing	7
4.	Gestione della Conversazione	17
	Buffer e Storico della Conversazione	18
	Riformulazione delle Query e guardrail per domande fuori contesto	19
	Generazione delle Risposte	24
5.	Scelta dei modelli	25
	Modello di Generazione 1: Gemini 2.0 Flash	25
	Modello di Generazione 2: Command-r-plus-04-2024	25
	Modello di Embedding: BGE-m3	26
	Motore di Indicizzazione: FAISS con Max Inner Product	27
6.	Utilizzo del chatbot	28
7.	Valutazione e Risultati	29
	Metriche di Valutazione	29
	Log delle Interazioni	29
	Risultati Sperimentali	30
8.	Conclusioni	33

1. INTRODUZIONE

Il chatbot sviluppato in questo progetto rappresenta un assistente virtuale avanzato progettato per rispondere alle domande relative al corso di "Natural Language Processing and Large Language Models" (anno accademico 2024/2025). Questo strumento non solo fornisce informazioni dettagliate sui contenuti didattici, ma è anche in grado di gestire richieste generali, come i dettagli sui docenti, i libri consigliati e altre informazioni correlate al corso.

Una delle caratteristiche distintive del sistema è la sua capacità di riconoscere domande fuori contesto, segnalando all'utente quando una richiesta esula dal dominio di competenza del chatbot. Per garantire prestazioni elevate, il sistema adotta un approccio Retrieval-Augmented Generation (RAG), che combina la potenza generativa dei Large Language Models (LLM) con un meccanismo di recupero delle informazioni basato su una knowledge base strutturata.

L'architettura del chatbot è stata progettata per garantire:

- Precisione: grazie a un sistema di retrieval ottimizzato che recupera le informazioni più rilevanti dalla knowledge base.
- Coerenza: attraverso una gestione avanzata del contesto conversazionale.
- Scalabilità: sfruttando modelli di embedding e indicizzazione efficienti.
- Robustezza: con meccanismi di controllo per evitare risposte errate o fuori tema.

Il progetto mira a dimostrare come l'integrazione di tecnologie NLP all'avanguardia possa migliorare l'accesso alle informazioni didattiche, offrendo agli studenti uno strumento interattivo e affidabile per lo studio.

2. ARCHITETTURA DEL SISTEMA

L'architettura del chatbot è stata progettata per garantire un flusso di elaborazione efficiente, dalla ricezione della domanda dell'utente alla generazione di una risposta accurata e contestualizzata. Il sistema si compone di diversi moduli interconnessi, ognuno con un ruolo specifico nel processo di elaborazione.

Componenti Principali

1. Preprocessing dei Dati

- **Estrazione del testo:** elaborazione di PDF (libri, slide) tramite librerie come PyPDF2 e OCR (Pytesseract) per estrarre contenuti testuali anche da immagini.
- **Pulizia e formattazione:** rimozione di duplicati, correzione di formattazione incoerente e ottimizzazione della leggibilità tramite modelli generativi (es. Gemini 2.0 Flash).
- **Arricchimento semantico:** integrazione di informazioni aggiuntive dal libro di testo per migliorare la completezza dei contenuti.

2. Sistema di Embedding e Indicizzazione

- **Modello di embedding:** utilizzo di **BGE-M3** per convertire il testo in rappresentazioni vettoriali ad alta dimensionalità, ottimizzando la similarità semantica.
- **Indice FAISS:** archiviazione efficiente degli embedding con ricerca Approximate Nearest Neighbor (ANN) basata su **Max Inner Product (MIP)**, che garantisce velocità e precisione nel retrieval.

3. Gestione della Conversazione

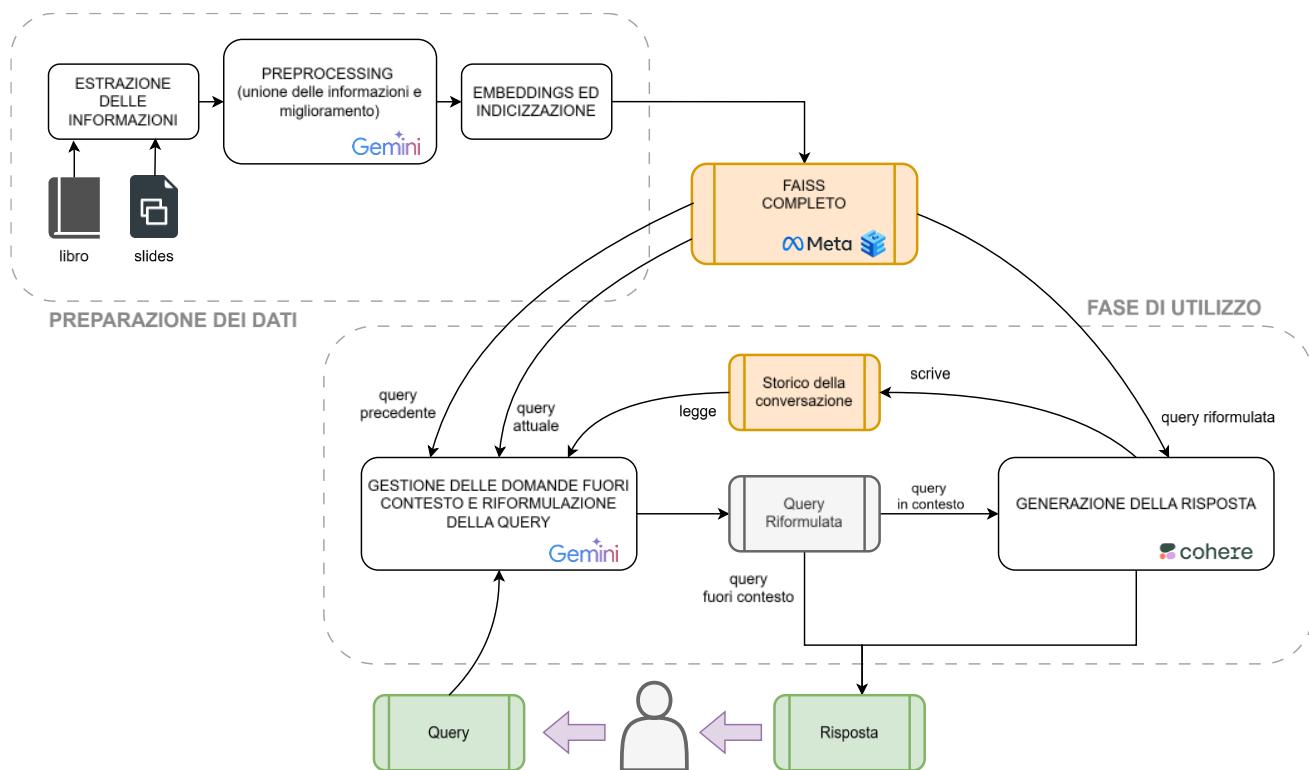
- **Buffer di contesto:** memorizzazione dello storico delle interazioni per mantenere la coerenza in domande di follow-up.
- **Query Reformulation:** riformulazione intelligente delle domande tramite Gemini 2.0 Flash, con riconoscimento di richieste ambigue o fuori contesto.

4. Retriever e Generatore di Risposte

- **Retriever:** ricerca dei chunk più rilevanti nella knowledge base in base alla query riformulata.
- **Generatore (Command-R-Plus):** produzione di risposte fluide e ben strutturate, integrate con i documenti recuperati.

Workflow del Sistema

- Ricezione della domanda:** L'utente invia una query tramite l'interfaccia web o un notebook di test.
- Riformulazione della query:**
 - Analisi del contesto conversazionale.
 - Verifica della pertinenza (domanda in-context vs. out-of-context).
 - Ottimizzazione della query per il retrieval.
- Recupero delle informazioni:**
 - Ricerca semantica nell'indice FAISS.
 - Selezione dei documenti più rilevanti.
- Generazione della risposta:**
 - Integrazione dei documenti recuperati nel prompt del generatore.
 - Produzione di una risposta naturale e ben formattata.
- Aggiornamento del contesto:**
 - Memorizzazione della domanda e della risposta nello storico.
 - Limitazione della memoria a 5 interazioni (FIFO) per evitare sovraccarico.

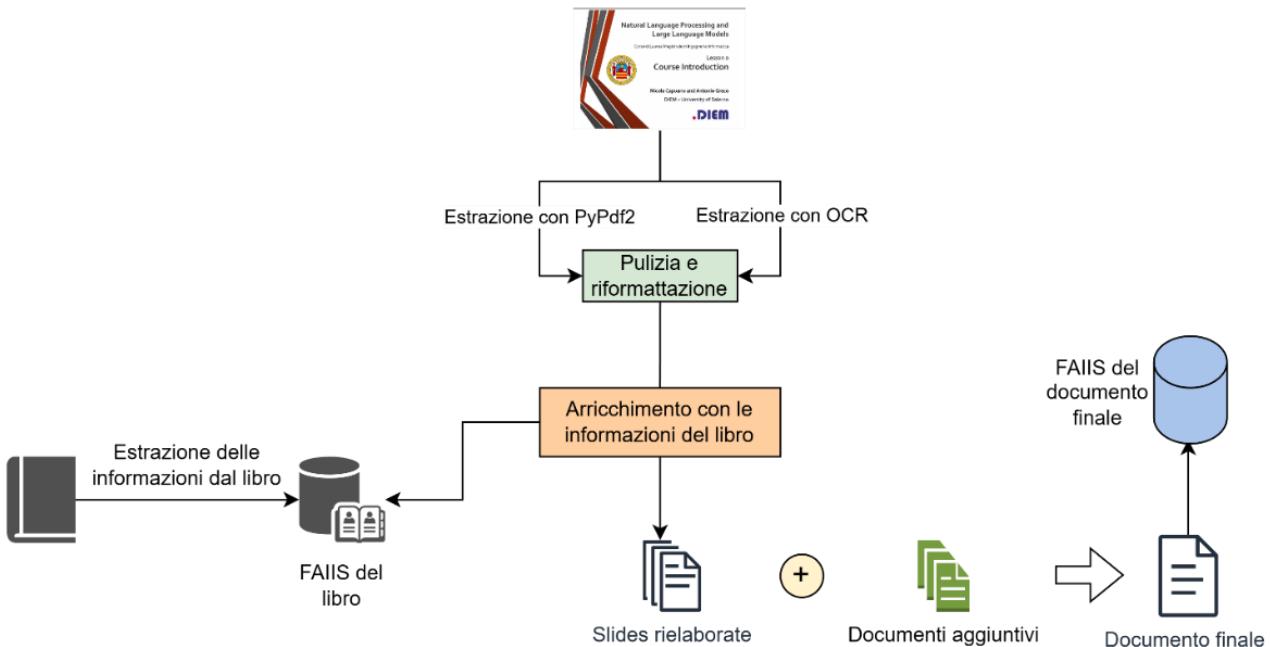


Tecnologie Chiave

Componente	Tecnologia/Modello	Ruolo
Embedding	BGE-M3	Conversione del testo in vettori ad alta dimensionalità.
Indicizzazione	FAISS (MIP)	Ricerca veloce dei documenti più rilevanti.
Riformulazione query	Gemini 2.0 Flash	Ottimizzazione delle domande e filtraggio di richieste fuori contesto.
Generazione risposte	Command-R-Plus	Produzione di risposte coerenti e dettagliate.
Estrazione testo	PyPDF2 + Pytesseract (OCR)	Elaborazione di PDF e immagini per il recupero del contenuto testuale.

3. PREPARAZIONE DEI DATI

La qualità del chatbot dipende fortemente dalla solidità della base di conoscenza. Il processo di preparazione dei dati è strutturato in diverse fasi, volte a garantire informazioni accurate e ben organizzate.



Fonti dei Dati

Le fonti utilizzate sono:

- **Libro di testo “Natural Language Processing IN ACTION” (H. Lane, C. Howard, H. M. Hapke)**
- **Slides del corso**
- **Sito web Unisa** (per prelevare la scheda del corso e i dati relativi ai docenti del corso)

Fasi del pre-processing

Il processo di preparazione dei dati si articola in diverse fasi, strutturate per garantire la massima qualità delle informazioni che andranno a costituire la base di conoscenza del chatbot.

1. Estrazione delle informazioni contenute nel libro

In questa fase è stato elaborato il PDF del libro, utilizzando la libreria **PyPDF2** per scorrere tutte le pagine e raccogliendone il contenuto testuale. Una volta completata l'estrazione, il testo viene scritto in un file .txt con codifica UTF-8, assicurando così la corretta conservazione dei caratteri.

2. Embeddings e indicizzazione del libro

Il testo precedentemente estratto dal libro è stato suddiviso in chunks utilizzando il **RecursiveCharacterTextSplitter**, settando `chunk_size` pari a 8000 e `chunk_overlap` a 150. Questa scelta è stata motivata dai seguenti fattori:

- Il `RecursiveCharacterTextSplitter` suddivide il testo seguendo la gerarchia del documento (paragrafi, sezioni, sottosezioni), evitando spezzature arbitrarie che potrebbero compromettere la comprensione semantica. Questo approccio è particolarmente utile per documenti accademici, che hanno una struttura ben definita.
- Un `chunk_size` di 8000 caratteri consente di mantenere una porzione ampia di testo in ogni frammento, preservando il contesto necessario la fase di RAG e riducendo il rischio di perdita di informazioni critiche. Valori inferiori avrebbero prodotto chunks troppo piccoli, frammentando maggiormente il contenuto e riducendo l'efficacia del retrieval.
- Il `chunk_overlap` di 150 caratteri è stato scelto per garantire una continuità tra i segmenti, riducendo il rischio che informazioni chiave vengano tagliate tra un chunk e l'altro. Un overlap più alto avrebbe comportato un aumento dello spazio occupato e dei costi computazionali, senza un beneficio significativo per il recupero delle informazioni.

Successivamente, per rappresentare il testo in uno spazio vettoriale, è stato utilizzato un modello open-source di Hugging Face per la generazione degli embeddings: **BGE-M3** di BAAI. Ogni chunk viene trasformato in un embedding che ne cattura il significato semantico. Infine, gli embeddings ottenuti vengono archiviati in un database **FAISS** (Facebook AI Similarity Search), che permette di eseguire ricerche rapide basate sulla similarità tra i vettori, ottimizzando così il recupero delle informazioni. In particolare, viene impiegata la strategia di similarità basata sul massimo prodotto interno (MAX_INNER_PRODUCT) per calcolare la somiglianza tra i vettori.

3. Estrazione delle informazioni contenute nelle slides (STEP_1)

Il testo contenuto nelle slides in formato PDF è stato estratto seguendo lo stesso approccio utilizzato per il libro. Tuttavia, il processo non è stato sufficiente per estrarre il testo presente all'interno delle immagini. Per affrontare questa limitazione, è stata impiegata la libreria **Pytesseract**, che utilizza il motore OCR (Optical Character Recognition) di Tesseract per riconoscere ed estrarre il testo dalle immagini contenute nel PDF, garantendo così l'accurata estrazione anche dei contenuti visibili come immagini.

È stato scelto un valore di 300 DPI (Dots Per Inch) per l'analisi OCR dei PDF poiché rappresenta il miglior compromesso tra qualità e prestazioni. Questa risoluzione garantisce un riconoscimento accurato del testo, evitando errori dovuti a immagini poco definite,

senza appesantire eccessivamente i file o rallentare l'elaborazione. Inoltre, 300 DPI è lo standard raccomandato per garantire contorni chiari e una separazione ottimale dei caratteri. Risoluzioni inferiori riducono la precisione, mentre valori superiori aumentano il peso dei file senza un miglioramento proporzionale.

Di seguito sono riportati esempi di estrazione delle informazioni.

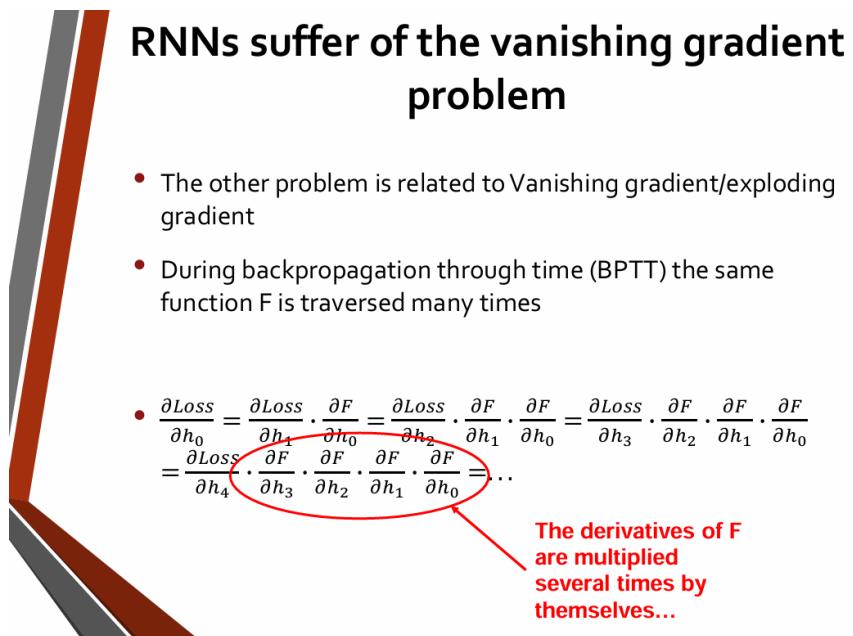


Figura 1: Slide originale

RNNs suffer of the vanishing gradient problem

- The other problem is related to Vanishing gradient/exploding gradient
- During backpropagation through time (BPTT) the same function F is traversed many times
- $\frac{\partial \text{Loss}}{\partial h_0}$
- $= \frac{\partial \text{Loss}}{\partial h_1}$
- $\cdot \frac{\partial F}{\partial h_0}$
- $= \frac{\partial \text{Loss}}{\partial h_2}$
- $\cdot \frac{\partial F}{\partial h_1}$
- $\cdot \frac{\partial F}{\partial h_0}$

RNNs suffer of the vanishing gradient problem

- The other problem is related to Vanishing gradient/exploding gradient
- * During backpropagation through time (BPTT) the same function F is traversed many times

e OLOSS OLOSS _ OF __ OLOSS OF OF _ OLOSS OF OF OF
a —ahoh_ oho_ ohz ohy oh, AN
OF @F OF OF
“oh3 ohy oh, oho J

The derivatives of F are multiplied several times by themselves...

Figura 2: Testo estratto con PyPdf2

Figura 3: Testo estratto con OCR

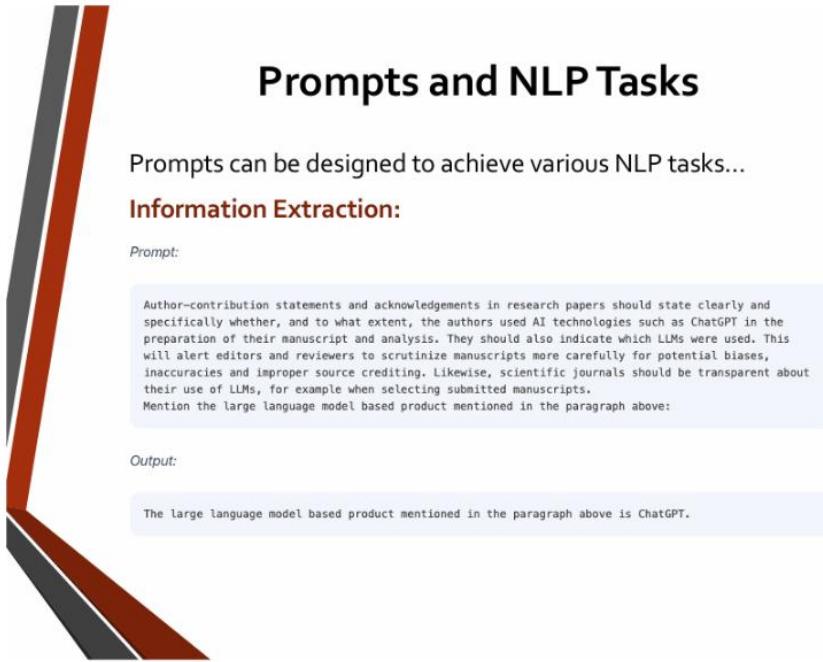


Figura 4: Slide n°2 originale

Prompts and NLP TasksPrompts can be designed to achieve various NLP tasks...Information Extraction:

Prompts and NLP TasksPrompts can be designed to achieve various NLP tasks...Question Answering:
Prompts and NLP TasksPrompts can be designed to achieve various NLP tasks...Text Classification:

Figura 5: Testo n°2 estratto con PyPdf2

Prompt:

Author-contribution statements and acknowledgements in research papers should state clearly and specifically whether, and to what extent, the authors used AI technologies such as ChatGPT in the preparation of their manuscript and analysis. They should also indicate which LLMs were used. This will alert editors and reviewers to scrutinize manuscripts more carefully for potential biases, inaccuracies and improper source crediting. Likewise, scientific journals should be transparent about their use of LLMs, for example when selecting submitted manuscripts.

Mention the large language model based product mentioned in the paragraph above:

Output:

The large language model based product mentioned in the paragraph above is ChatGPT.

Figura 6: Testo n°2 estratto con OCR

I risultati mostrati evidenziano come l'approccio di estrazione del testo vari a seconda della natura del contenuto presente nelle slide. Nei casi in cui il testo è chiaramente visibile e non è associato a elementi grafici complessi, l'estrazione diretta del testo dal PDF funziona in modo ottimale, garantendo una qualità elevata del contenuto estratto, senza richiedere risorse aggiuntive per il processo di OCR. D'altra parte, in presenza di immagini contenenti testo, come nelle slide che includono diagrammi, tabelle o screenshot, l'approccio OCR con Pytesseract si rivela fondamentale. Questo metodo permette di riconoscere e estrarre il testo presente nelle immagini, anche se non è sempre perfetto. In questi casi, il testo estratto

tramite OCR può presentare piccole imprecisioni, soprattutto se la qualità dell'immagine non è ottimale, ma rappresenta comunque una soluzione efficace per ottenere contenuti altrimenti inaccessibili. La combinazione dei due metodi — estrazione del testo diretto per il contenuto testuale e OCR per le immagini — consente di ottenere una copertura completa delle informazioni contenute nelle slide, sfruttando i punti di forza di ciascun approccio in modo complementare.

4. Pulizia e riformattazione delle informazioni estratte dalle slides (STEP_2)

Dopo aver unito le informazioni estratte dalle slides in formato PDF e in formato immagine con OCR, la funzione `generate_better_text_of_slide` si occupa della pulizia e riformattazione del testo, al fine di rimuovere informazioni duplicate, migliorare la leggibilità, correggere problemi di formattazione, spaziature irregolari e interruzioni di riga. In questo modo si ottiene un'elaborazione uniforme e un miglioramento complessivo della qualità del testo estratto.

Per questa operazione viene utilizzato un modello generativo (**gemini-2.0-flash**) che riorganizza il testo mantenendone intatto il contenuto informativo, utilizzando il seguente prompt:

”

The following text has been extracted from a PDF and is poorly formatted, with inconsistent spacing, line breaks and structure.

Your task is to rewrite the text to improve its readability and formatting.

Specifically:\n\n

1. Remove unnecessary line breaks and spaces to create a smooth, continuous flow of text.\n
2. Correct any formatting issues, such as misplaced punctuation, inconsistent capitalization, or fragmented sentences.\n
3. Ensure the text is clean and easy to read, with proper spacing and structure.\n
4. Is important that you don't lose any information!.\n
5. When formulas or code pieces are recognized, rewrite them better using your knowledge.\n
6. Avoid to use or recite any copyrighted content.\n
7. Do not add unnecessary information, such as 'here is a reformatted text'.\n

Here is the text to reformat:\n\n

{text}

”

```
A magnitude 6.7 earthquake rattled Papua New Guinea  
early Friday afternoon, according to the U.S. Geological  
Survey. The quake was centered about 200 miles  
north-northeast of Port Moresby and had a depth of 28  
miles. No tsunami warning was issued...
```

```
<Article 1  
summary>
```

```
NLP task behind this app: Find a model for this task:
```

```
Summarization
```

```
Hugging Face Hub > 176,620 models.  
Filter by task > 960 models.
```

```
Then...2 Consider your needs.
```

```
Extractive: Select representative pieces of text.
```

```
A
```

```
stractive: Generate new text.
```

Figura 7: Testo prima della pulizia e riformattazione

```
A magnitude 6.7 earthquake rattled Papua New Guinea early Friday afternoon, according to the U.S. Geological Survey.  
NLP task behind this app: Summarization
```

```
Find a model for this task: Hugging Face Hub > 176,620 models.  
Filter by task > 960 models.
```

```
Then...2 Consider your needs.
```

```
Extractive: Select representative pieces of text.
```

```
Abstractive: Generate new text.
```

Figura 8: Testo dopo la pulizia e riformattazione

- T5 (Text-to-Text Transfer Transformer) is a language model based on an encoder-decoder transformer developed by Google Research.

- T5 comes in multiple sizes to suit different resource constraints:

T5 Encoder Attention Decoder Embedding
Version Blocks Heads Blocks Dimensionality
T5-Small 6 8 6 512

T5-Base 12 12 12 768

T5-Large 24 16 24 1024

T5-XL 24 32 24 2048

T5-XXL 24 64 24 4096

T5 input encoding

- T5 uses a SentencePiece tokenizer with a custom vocabulary for its input encoding.

Figura 9: Testo n°2 prima della pulizia e riformattazione

```
### T5  
T5 (Text-to-Text Transfer Transformer) is a language model based on an encoder-decoder transformer developed by Google Research.
```

```
T5 comes in multiple sizes to suit different resource constraints:
```

Version	Encoder Blocks	Attention Heads	Decoder Blocks	Embedding Dimensionality
T5-Small	6	8	6	512
T5-Base	12	12	12	768
T5-Large	24	16	24	1024
T5-XL	24	32	24	2048
T5-XXL	24	64	24	4096

```
### T5 Input Encoding  
T5 uses a SentencePiece tokenizer with a custom vocabulary for its input encoding.
```

Figura 10: Testo n°2 dopo la pulizia e riformattazione

5. Arricchimento dei contenuti estratti dalle slides con ulteriori informazioni (STEP 3)

Le informazioni estratte dalle slides vengono arricchite con ulteriori dettagli tratti dal libro di testo, migliorando così la completezza e la profondità del contenuto: quando viene elaborata una slide, il suo contenuto viene convertito in embeddings e confrontato con quelli presenti nell'indice FAISS ottenuto dal libro. Questo permette di individuare rapidamente i passaggi del libro che condividono una forte affinità concettuale con il testo della slide. Le informazioni recuperate vengono quindi integrate nel contenuto originale, mantenendone la struttura e il significato, ma arricchendolo con dettagli aggiuntivi che migliorano la chiarezza, la coerenza e la completezza dell'esposizione.

Prompt utilizzato:

””

Task Description

Enhance the given text by preserving all its original information while improving clarity, coherence, and depth. Expand on key concepts by integrating relevant insights and additional context without altering the meaning or omitting any details. Ensure that the enhanced text flows naturally and remains logically structured.

Provided Information

Original Text:

{text}

Additional Context:

{additional_info}

Guidelines & Constraints

Retain all information from the original text without omitting any details.

Add relevant explanations and context to enrich understanding.

Improve readability, coherence, and logical flow.

Do not introduce personal opinions or unverifiable information.

Maintain a structured format with sections separated by the delimiter:

<-----section----->

Keep the dimension of each section under 6000-7000 characters.

Verification Step

After generating the enhanced text, perform a rigorous self-check to ensure that no information from the original text has been lost or misrepresented. Compare the enhanced version with the original text and confirm:

That all key points and details are present.

That nothing has been omitted, reinterpreted incorrectly, or altered in meaning.

If any information is missing or distorted, refine the output accordingly before finalizing it.

””

A seguire alcuni esempi dei risultati ottenuti:

```
*  **Single-label:** Assigns each document in D to only one class in C.
*  **Binary:** Like Single-label but C has only two classes. Classification is a decision between a class and its complement.
*  **Multi-label:** Assigns each document to a variable number of classes in C. Can be reduced to a series of binary decisions.

**ML-Based Classification**

* A machine learning model is trained on a set of annotated text documents.
* Each document in the training set is associated with one or more class labels.
* After training, the model can predict the category (or categories) for a new document.
* The classifier may provide a confidence measure.
* A vector representation of documents, such as TF-IDF, must be used.

**Topic Labelling Example**

**Classifying Reuters News**

The Reuters 21578 dataset is multi-class and multi-label:
```

Figura 11: Testo prima dell'arricchimento con le informazioni del libro e divisione in sezioni

```
*  **Single-label:** Each document in the set D is assigned to *only one* class from the set C. For example, a news article can only be categorized
*  **Binary:** This is a special case of single-label classification where the set C contains only two classes. This is a decision between a class
*  **Multi-label:** Each document can be assigned to *one or more* classes from the set C. For example, a movie can be labeled as 'action', 'comedy'

<-----section----->

**ML-Based Classification**

Machine learning (ML) provides powerful techniques for automating text classification. The general process involves:

1. **Training Data:** A machine learning model is trained on a set of annotated text documents. These annotated documents form the training set.
2. **Labeling:** Each document in the training set is associated with one or more class labels, indicating its category or categories.
3. **Model Training:** The machine learning algorithm learns patterns and relationships between the text content and the assigned labels during the
4. **Prediction:** After training, the model can predict the category (or categories) for a new, unseen document.
5. **Confidence Measure:** The classifier may also provide a confidence score or probability, indicating how certain it is about its prediction.
6. **Vector Representation:** A crucial step is converting text documents into numerical vectors that machine learning models can process. Common techniques include:
    * **TF-IDF (Term Frequency-Inverse Document Frequency):** Weighs words based on their frequency within a document and their rarity across the
    * **Word Embeddings (Word2Vec, GloVe, FastText):** Represents words as dense vectors in a high-dimensional space, capturing semantic relations
    * **BERT, transformer-based encoders:** Create contextualized word embeddings to better represent the semantic meaning of text

<-----section----->

**Topic Labeling Example**

**Classifying Reuters News**

The Reuters-21578 dataset is a widely used benchmark dataset for text classification. It consists of news articles from Reuters newswire, categorized into over 200 distinct topics.
```

Figura 12: Testo dopo l'arricchimento con le informazioni del libro e divisione in sezioni usando il separatore <-----section----->

```
**Model selection**

Also consider:
    Search for examples and datasets not just models.

Is the model "good" at everything, or was it fine-tuned for a specific task?
    Which datasets were used for pre-training and/or fine-tuning?
    Ultimately, it's about your data and users.

* Define KPIs.
* Test on your data or users.
```

Figura 13: Testo n°2 prima dell'arricchimento con le informazioni del libro e divisione in sezioni

```
**Model selection**

Other considerations:
    * **Search for examples and datasets:** Look for resources that demonstrate the model's performance and suitability for your specific use case.
    * **Model specialization:** Determine whether the model is generally capable or fine-tuned for a specific task.
    * **Training data:** Identify the datasets used for pre-training and/or fine-tuning, as this influences the model's bias and performance.

Ultimately, model selection should be data and user-driven:
    * **Define KPIs:** Establish key performance indicators to measure the model's effectiveness.
    * **Test on your data or users:** Evaluate the model's performance on your specific dataset and with your target users to ensure it meets your needs.
```

Figura 14: Testo n°2 dopo l'arricchimento con le informazioni del libro e divisione in sezioni usando il separatore <-----section----->

6. Creazione del file finale

Il risultato della fase precedente è disponibile nella cartella [data/preprocessing/STEP3](#). Qui si trova un file finale per ciascun pacchetto di slide di partenza, opportunamente preprocessato, arricchito con le informazioni del libro e suddiviso in sezioni.

In questa fase, tutte le informazioni rielaborate vengono unite in un unico file contenente l'intero contenuto del corso.

Le varie slide vengono separate utilizzando il delimitatore `<-----section----->`.

Inoltre, le informazioni aggiuntive relative al corso, ai docenti assegnati e ad altri dettagli rilevanti, non presenti né nel libro né nelle slide, vengono integrate in questo file senza ulteriori trasformazioni. Anche queste informazioni vengono inserite utilizzando lo stesso separatore impiegato per le slide, garantendo così uniformità nella struttura del documento finale.

➤ STEP_1
➤ STEP_2
➤ STEP_3
└ 0. Course Introduction.txt
└ 1. NLP Overview.txt
└ 2. Representing Text.txt
└ 3. Math with Words.txt
└ 4. Text Classification.txt
└ 5. Word Embeddings.txt
└ 6. Neural Networks for NLP.txt
└ 7. Dialog Engines.txt
└ 8. Building a Chatbot.txt
└ 09. Transformers I.txt
└ 10. Transformers II.txt
└ 11. From Transformers to LLMs.txt
└ 12. HuggingFace.txt
└ 13. Encoder-only Transformers.txt
└ 14. Decoder-only Transformers.txt
└ 15. Encoder-Decoder Transformers.txt
└ 16. Final Project.txt
└ 17. Fine tuning.txt
└ 18. Prompt Engineering.txt
└ 20. RAG.txt
└ 21. RLHF for LLMs.txt
└ 22. Guardrails for LLMs.txt

Figura 15: directory STEP 3

In questo modo si è ottenuto **un unico file testuale che contiene tutte le informazioni** che costituiranno la conoscenza di base del chatbot.

7. Embeddings e indicizzazione del file finale

Il testo finale ottenuto viene quindi suddiviso automaticamente dall'LLM in sezioni logiche delimitate dai marcatori specifici utilizzati (`<-----section----->`).

È stato scelto questo metodo di suddivisione, il **Section Based Chunking**, in quanto presenta numerosi vantaggi rispetto ai metodi di chunking più tradizionali come il Fixed Size Chunking o il Recursive Chunking. Tra questi abbiamo:

- **Maggiore coerenza semantica**, poiché i chunks vengono creati seguendo la struttura logica del documento ed evitando di spezzare concetti o frasi a metà, migliorando di conseguenza il recupero delle informazioni e la gestione di documenti complessi e strutturati
- **Maggiore efficienza nell'uso dei token**, in quanto dividendo concettualmente i documenti non è più necessario un overlap tra le diverse sezioni, evitando così la ripetizione di contesto e riducendo il costo computazionale

Queste sezioni così ricavate vengono convertite in rappresentazioni vettoriali (embeddings) utilizzando il modello di embedding BGE-M3. Gli embeddings vengono infine indicizzati tramite FAISS.

Dopo aver generato i chunks, ci è chiesto se la loro diversa dimensione potesse influenzare la fase di retrieval. Per valutare questo aspetto, sono state effettuate **956 retrieval con 956 query**, contenute nel file “[LLM\data\questions\6Marzo2025_ALL.json](#)”.

Successivamente, è stata analizzata la correlazione tra la dimensione dei chunks e il numero di volte in cui questi venivano recuperati.

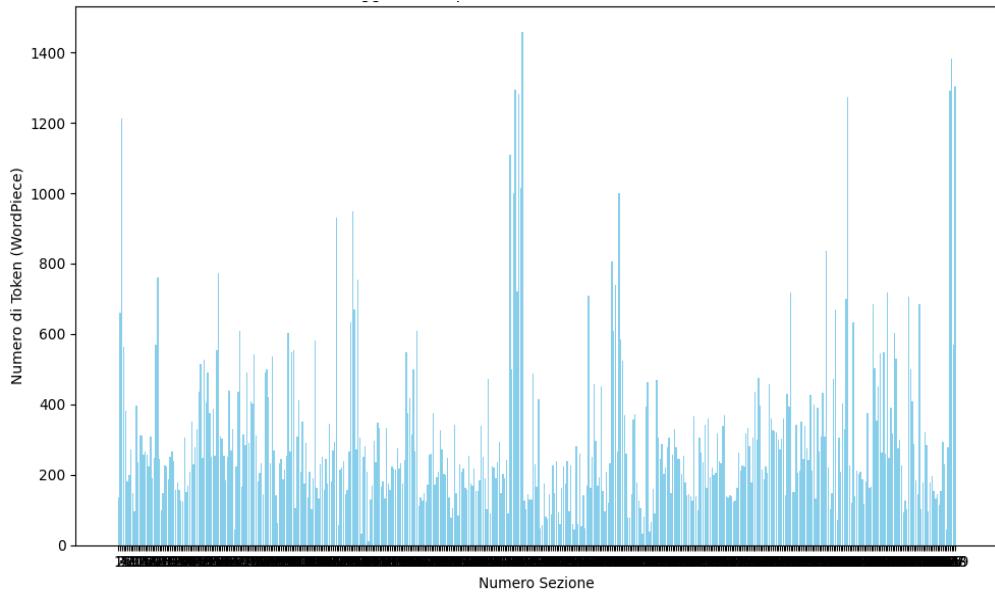


Figura 16: dimensione di ogni chunk

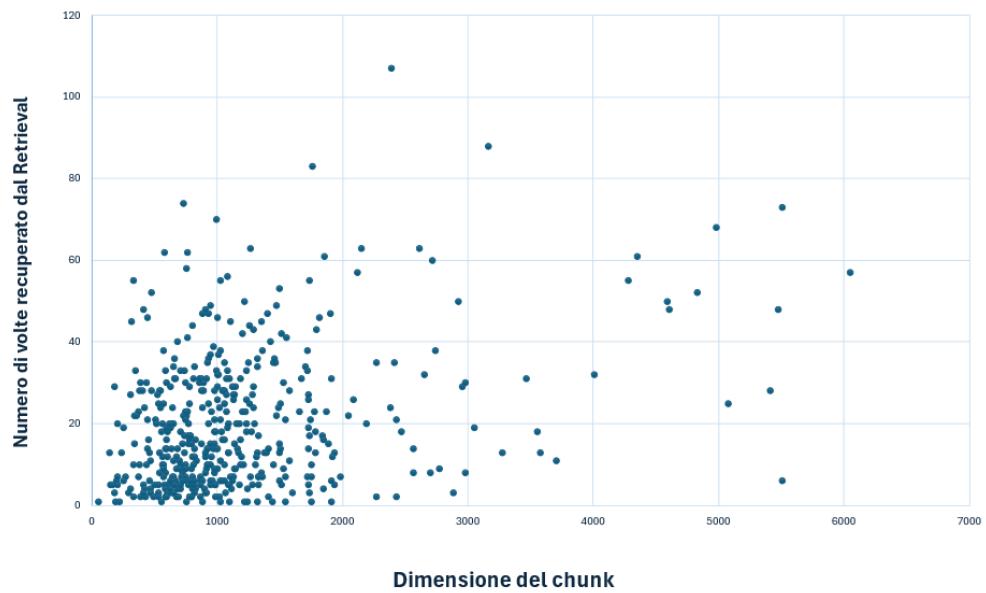


Figura 17: Scatter plot dimensione del chunk – numero di volte in cui è stato recuperato

Il grafico mostra la distribuzione del numero di recuperi in funzione della dimensione del chunk: dallo scatter plot non è evidente un chiaro trend di crescita o decrescita netta e non sembra emergere una correlazione evidente tra dimensione e frequenza di retrieval.

- **Coefficiente di Correlazione $\approx 0,335$**

4. GESTIONE DELLA CONVERSAZIONE

Il processo di generazione delle risposte comprende:

1. Recupero del contesto:

- Viene eseguito un recupero dei documenti rilevanti per la domanda corrente (retrieved_docs_current), e il loro contenuto viene concatenato in una stringa (retrieved_context).
- Se c'è una domanda precedente nella cronologia (self.history), viene recuperata la query riformulata e si ottengono i documenti rilevanti per essa. Questo contesto viene aggiunto al contesto attuale.

2. Riformulazione della query:

- La query originale, insieme alla cronologia della conversazione e al contesto recuperato, viene inviata a un sistema di riformulazione delle domande (query_handling_chain) per ottimizzare la query in modo che possa ottenere una risposta migliore dal modello di QA (domanda-risposta).

3. Verifica della pertinenza della query:

- Se la query riformulata contiene un segnale di "fuori contesto" (stringa "OUT_OF_CONTEXT_QUESTION"), viene invocato un processo di chiarimento per gestire la situazione, e non vengono recuperati ulteriori documenti.
- Se la query è valida, viene utilizzata la catena di QA (qa_chain) per ottenere una risposta alla query riformulata e i documenti sorgente associati.

4. Registrazione dell'interazione:

- I dettagli dell'interazione (domanda, query riformulata, risposta, documenti recuperati) vengono registrati nel log delle interazioni.
- La nuova domanda, la risposta e la query riformulata vengono aggiunti alla cronologia per future interazioni.

5. Restituzione del risultato:

- La funzione restituisce un oggetto JSON che contiene i dettagli dell'interazione, come la domanda, la risposta e i documenti di origine.

```

def ask(self, question: str) -> dict:
    retrieved_docs_current = self.retriever.get_relevant_documents(question)
    retrieved_context = "\n".join([doc.page_content for doc in retrieved_docs_current])

    previous_query = self.history[-1]['reformulated_query'] if self.history else ""
    retrieved_docs_previous = self.retriever.get_relevant_documents(previous_query) if previous_query else []
    retrieved_context += "\n" + "\n".join([doc.page_content for doc in retrieved_docs_previous])

    reformulated_query = self.query_handling_chain.invoke({
        "question": question,
        "chat_history": self.get_history_for_prompt(),
        "retrieved_context": retrieved_context
    })["text"].strip()

    if "OUT_OF_CONTEXT_QUESTION" in reformulated_query:
        response = self.clarify_response(reformulated_query)
        retrieved_docs = []
    else:
        result = self.qa_chain.invoke({"question": reformulated_query})
        response = result["answer"]
        retrieved_docs = result.get("source_documents", [])

    interaction_json = self.log_interaction(question, reformulated_query, response, retrieved_docs)
    self.add_to_history(question, response, reformulated_query)
    return interaction_json

```

Un aspetto critico del sistema è la gestione dinamica della conversazione, essenziale per mantenere coerenza e continuità nelle interazioni. Il sistema implementa meccanismi avanzati per memorizzare, analizzare e utilizzare lo storico delle conversazioni per generare risposte contestualizzate.

Buffer e Storico della Conversazione

- **Buffer di Conversazione:**

Il sistema memorizza lo storico delle interazioni mediante un dizionario **memory_history** che mantiene traccia del dialogo. Questo componente è essenziale per permettere al sistema di mantenere il contesto attraverso più scambi comunicativi.

- **Gestione dello Storico:**

Lo storico viene limitato a un massimo di `max_history` interazioni per evitare sovraccarichi di contesto. Quando il limite viene raggiunto, le interazioni più vecchie vengono rimosse secondo un meccanismo FIFO (First In, First Out).

Dopo diverse prove il valore scelto per il massimo numero di interazioni da salvare è 5.

- **Struttura del Dato:**

Ogni interazione è memorizzata come un dizionario contenente:

- domanda originale dell'utente
- query riformulata
- risposta finale fornita

- **Formattazione dello Storico:**

Per l'analisi, lo storico viene formattato come una sequenza di coppie `User: [...]` `Assistant: [...]`, facilitando l'interpretazione da parte del modello di language generation.

Riformulazione delle Query e guardrail per domande fuori contesto

Per ottimizzare il recupero delle informazioni, il sistema implementa una catena di riformulazione delle query che tramite **Gemini 2.0 Flash** trasforma la domanda dell'utente in una query ottimizzata per la ricerca nella knowledge base.

Prompt di Riformulazione

Il sistema utilizza il seguente prompt per riformulare le query:

...

```
You are an expert assistant specialized in NLP and LLMs.  
Analyze the user's query considering conversation history and retrieved documents.  
Determine if the question is a follow-up or independent and if it belongs to the  
LLMs/NLP domain.
```

****Steps:****

1. ****Identify Follow-up Queries:**** If the question depends on prior context, merge relevant details into a self-contained query.
2. ****Analyze Retrieved Documents:**** Use the retrieved context to determine if the question is relevant.
3. ****Determine Domain Relevance:**** If the question is unrelated to LLMs/NLP, return:
OUT_OF_CONTEXT_QUESTION: " say that you can't response, for example saying that the question is outside your knowledge domain."
4. ****Format Output:****
 - If valid, return only the refined query.
 - If out of context, return "OUT_OF_CONTEXT_QUESTION: <response>".

****Few-shot Examples:****

****Example 1 - Follow-up Query:****

****Conversation History:****

User: How does GPT-4 handle long context?

Assistant: GPT-4 uses positional embeddings and attention mechanisms to process long contexts efficiently.

User: Can it handle 10,000 tokens?

****Rewritten Query:**** "Can GPT-4 handle 10,000 tokens given its architecture?"

****Example 2 - Independent Query:****

****User Question:**** "What is zero-shot learning in NLP?"

****Rewritten Query:**** "What is zero-shot learning in NLP?"

****Example 3 - Out of Context Query:****

****User Question:**** "Who won the FIFA World Cup in 2018?"

****Response:**** "OUT_OF_CONTEXT_QUESTION: I'm sorry but I can't provide information about football. I specialize in answering questions related to NLP and LLMs."

****Example 4 - Out of Context Query:****

****User Question:**** "What is the capital of Japan?"

****Response:**** "OUT_OF_CONTEXT_QUESTION: I'm designed to assist with NLP and LLM-related topics. General knowledge queries are outside my domain."

****Example 5 - Out of Context Query:****

****User Question:**** "Can you recommend a good sci-fi book?"

```

**Response:** "OUT_OF_CONTEXT_QUESTION: My expertise is in NLP and LLMs. I
recommend checking book review platforms for such recommendations."

**Example 6 - Out of Context Query:**
**User Question:** "Who invented the light bulb?"
**Response:** "OUT_OF_CONTEXT_QUESTION: This question falls outside my
specialized domain of NLP and LLMs."

**Example 7 - Are you sure Query:**
**Conversation History:**
User: How does GPT-4 handle long context?
Assistant: GPT-4 uses positional embeddings and attention mechanisms to
process long contexts efficiently.
**User Question:** "Are you sure"
**Response:** "OUT_OF_CONTEXT_QUESTION: Yes, i'm sure. GPT-4 handle long
context using positional embeddings.

**Conversation History:**
{chat_history}
**User Question:**
{question}
**Retrieved Context:**
{retrieved_context}
...

```

Il prompt fornito combina diverse strategie per guidare il comportamento del modello in modo efficace e controllato. In particolare, si possono individuare quattro principali categorie di prompting utilizzate:

- **Few-shot Prompting**

Il prompt include esempi specifici di input e output attesi per guidare il modello. I "few-shot examples" mostrano come distinguere tra:

- Follow-up query (esempio 1)
- Domanda indipendente (esempio 2)
- Domanda fuori contesto (esempi 3-6)
- Conferma di una risposta precedente (esempio 7)

L'uso di esempi aiuta il modello a generalizzare il comportamento desiderato senza dover esplicitare ogni possibile casistica nelle istruzioni iniziali.

- **Meta Prompt**

Il prompt stabilisce come il modello deve ragionare e prendere decisioni. Non si limita a chiedere una risposta, ma gli impone di:

- Analizzare la query in base alla cronologia della conversazione.
- Controllare il contesto recuperato.
- Determinare se la domanda è pertinente o fuori ambito.
- Riformulare le query di follow-up in modo compatto, integrando il contesto necessario.

- Role Prompting

All'inizio del prompt, il modello viene definito come un esperto di NLP e LLMs. Questo approccio serve a:

- Limitare il comportamento del modello al dominio specifico di competenza.
- Prevenire risposte fuori tema, migliorando la qualità delle interazioni.

L'assegnazione di un ruolo chiaro aiuta a focalizzare il modello su un contesto preciso, riducendo il rischio di risposte generiche o imprecise.

- Structured Prompting

Il prompt è organizzato in modo chiaro e sistematico, facilitando un comportamento prevedibile e coerente. La struttura segue una logica ben definita:

- Descrizione del ruolo e del compito ("You are an expert assistant specialized in NLP and LLMs...").
- Elenco dei passaggi che il modello deve seguire per analizzare la domanda.
- Formato di output stabilito, con due possibilità:
 - Una query riformulata, se la domanda è pertinente.
 - Un messaggio standardizzato (OUT_OF_CONTEXT_QUESTION), se la domanda è fuori contesto.
- Esempi strutturati, che coprono diverse tipologie di input utente.

Questa organizzazione permette di guidare il modello in modo preciso e ridurre l'ambiguità nelle risposte.

La funzione principale di questo prompt è analizzare la domanda dell'utente tenendo conto della cronologia della conversazione e dei documenti recuperati, al fine di determinare se la richiesta è un follow-up di una domanda precedente o un'interrogazione indipendente. Inoltre, valuta la pertinenza del quesito rispetto al dominio degli LLM e dell'NLP, restituendo una risposta appropriata.

```
def ask(self, question: str) -> dict:
    retrieved_docs_current = self.retriever.get_relevant_documents(question)
    retrieved_context = "\n".join([doc.page_content for doc in retrieved_docs_current])

    previous_query = self.history[-1]['reformulated_query'] if self.history else ""
    retrieved_docs_previous = self.retriever.get_relevant_documents(previous_query) if previous_query else []
    retrieved_context += "\n" + "\n".join([doc.page_content for doc in retrieved_docs_previous])

    reformulated_query = self.query_handling_chain.invoke({
        "question": question,
        "chat_history": self.get_history_for_prompt(),
        "retrieved_context": retrieved_context
    })["text"].strip()
```

In questo caso, si osserva che il contesto estratto risulta rilevante sia per la query attuale che per quella precedente. Questa scelta è motivata dalla necessità di risolvere un problema specifico: qualora vi sia una domanda di follow-up che non contenga esplicitamente il

contesto, come ad esempio “go more in detail about it”, effettuare un recupero dei documenti basato solo sulla similarità con tale query non porterebbe a risultati significativi. Di conseguenza, il sistema avrebbe un’alta probabilità di classificare la query come fuori contesto. Includendo anche le informazioni estratte dalla domanda precedente, si riesce a risolvere questa problematica, migliorando così la pertinenza dei risultati.

Se la domanda è fuori contesto, il sistema genera un messaggio che informa l’utente dell’assenza di una risposta pertinente. Il prompt fornisce anche esempi strutturati per illustrare il suo funzionamento, evidenziando casi di riformulazione di query di follow-up, domande indipendenti e domande fuori dominio. L’integrazione di questa logica nel flusso di elaborazione consente al sistema di migliorare la coerenza delle risposte e garantire una gestione più efficace delle interazioni utente-modello.

La riformulazione della query consente di gestire in modo corretto sia domande scritte male che domande di follow-up, di seguito diversi test eseguiti.

1. Domanda scritta male

can u talk me abut faiss indecs?

REFORMULATED QUERY:
Can you tell me about FAISS indexes?

FAISS Indexes Explained

FAISS (Facebook AI Similarity Search) is a library designed for efficient similarity search and clustering of high-dimensional vectors. It's particularly useful in applications like information retrieval, image recognition, and recommendation systems where finding similar vectors quickly is crucial.

Key Concepts:

2. Follow-up question

is TF-IDF an argoument of this course?

REFORMULATED QUERY:
"Is TF-IDF discussed in this course?"

Yes, TF-IDF is discussed in this course. The course covers the following aspects of TF-IDF:

- Limitations of TF-IDF in capturing semantic relationships between words.
- The concept of TF-IDF as a weighting scheme.
- Applications of TF-IDF, such as information retrieval, information filtering, and text classification.
- The components of TF-IDF: Term Frequency (TF) and Inverse Document Frequency (IDF).
- The formula for calculating Inverse Document Frequency (IDF).

go more in details

REFORMULATED QUERY:
Based on the conversation history and the provided documents, the user is asking for more details about a previously discussed topic. Since the last topic discussed was TF-IDF, the refined query is: "Can you provide more details about TF-IDF?"

TF-IDF (Term Frequency-Inverse Document Frequency) is a method used to represent text data by quantifying the importance of words within a document relative to a larger collection of documents (corpus). Here's a breakdown:

3. Altro esempio di Follow-up

who are the professors of this course?

REFORMULATED QUERY:

What are the names of the instructors for the Natural Language Processing and Large Language Models course?

The instructors for the Natural Language Processing and Large Language Models course are Nicola Capuano and Antonio Greco.

give me their resumes

REFORMULATED QUERY:

The question "give me their resumes" is a follow-up to the previous question "who are the professors of this course?". The professors of the course are Nicola Capuano and Antonio Greco. **Rewritten Query:** "Give me the resumes of Nicola Capuano and Antonio Greco."

Nicola Capuano

Nicola Capuano is an Associate Professor at the Department of Information and Electrical Engineering and Applied Mathematics (DIEM) at the University of Salerno. He obtained his degree in Computer Science and his Ph.D. in Computer Science and Computer Engineering from the University of Salerno. At the same university, he also held a four-year research fellowship on the topic of "Artificial Intelligence." Before pursuing an academic career, he collaborated with private research institutes, including the Center for Research in Pure and Applied Mathematics and the Center of Excellence for Software Technology. He also served as a researcher at the School of Engineering of the University of Basilicata. In 2021, he achieved the National

4. Domande fuori contesto e follow-up

is pope Francesco alive?

REFORMULATED QUERY:

OUT_OF_CONTEXT_QUESTION: I'm sorry, I cannot provide information about that, as my expertise lies in NLP and LLMs.

I'm sorry, I cannot provide information about that, as my expertise lies in NLP and LLMs.

are you sure?

REFORMULATED QUERY:

OUT_OF_CONTEXT_QUESTION: I'm designed to assist with NLP and LLM-related topics. I can't provide information about people.

I'm designed to assist with NLP and LLM-related topics. I can't provide information about people.

Generazione delle Risposte

Una volta riformulata la query, il sistema genera risposte utilizzando una catena di generazione RAG (Retrieval-Augmented Generation).

Prompt di Generazione

Il sistema utilizza il seguente prompt per generare risposte:

```
"""
You are an advanced AI assistant using Retrieval-Augmented Generation (RAG).
Use the provided knowledge to generate an accurate and coherent response.
if you lack sufficient context, say you cannot answer.

**Context:**
{context}

**User Question:**
{question}

**Response Format:**
- If you can answer, provide a well-structured response.
- If not, respond: "I'm sorry, but I don't have enough information to answer."
- Generation language: ENGLISH.
- !IMPORTANT: generate a response in HTML div format !.

"""

```

A differenza del primo prompt, che definisce una logica decisionale chiara e articolata per guidare il modello nell'analisi della query, questo secondo prompt adotta un approccio più essenziale. Non richiede una valutazione approfondita del contesto o una distinzione tra diverse tipologie di domande, ma si limita a orientare il modello verso la generazione dell'output, ponendo maggiore enfasi sulla sua forma piuttosto che sul processo di ragionamento sottostante, riducendo anche il carico computazionale.

5. SCELTA DEI MODELLI

Nel progetto, la selezione dei modelli è stata effettuata con l'obiettivo di ottimizzare il recupero delle informazioni e la generazione delle risposte nel chatbot. Ogni componente è stata scelta per garantire un equilibrio tra accuratezza, efficienza computazionale e scalabilità. Di seguito vengono giustificate le scelte dei modelli utilizzati.

Modello di Generazione 1: Gemini 2.0 Flash

Per la riformulazione della domanda, è stato scelto **Gemini 2.0 Flash** (temperatura: 0.5), in quanto offre un'eccellente combinazione di velocità ed efficienza. I motivi principali della scelta sono:

- **Qualità delle risposte:** in grado di fornire risposte coerenti e contestualmente rilevanti, integrando informazioni recuperate dal sistema RAG.
- **Ottimizzazione per scenari real-time:** rispetto ad altri modelli più pesanti, Gemini 2.0 Flash permette una generazione rapida senza sacrificare significativamente la qualità, essenziale per garantire un'esperienza utente fluida nel chatbot.
- **Alte capacità di adattarsi ad istruzioni data nel prompt.**
- **Utilizzo limitato gratuito**, adatto alle esigenze del progetto.
- **Utilizzo tramite API:** non è stato necessario fornirsi di un elaboratore in grado di eseguire il modello.

Modello di Generazione 2: Command-r-plus-04-2024

Per la seconda query, il cui compito principale era generare risposte basate sulle informazioni recuperate dal retrieval, è stato scelto il modello **command-r-plus-04-2024** (temperatura: 0.5). Questo modello, ottimizzato per attività di **Retrieval-Augmented Generation (RAG)**, ha dimostrato prestazioni eccellenti.

La selezione dei modelli è avvenuta attraverso un processo strutturato. Abbiamo utilizzato un **LLM** per generare circa **950 domande**, partendo da una versione precedente delle sezioni del testo. Il procedimento ha previsto l'invio di ciascuna sezione come prompt al modello, che ha restituito **cinque domande per volta**. Queste domande, salvate nel file *LLM\data\questions\6Marzo2025_ALL.json*, sono state successivamente impiegate per testare l'accuratezza delle risposte fornite dagli assistenti.

Per ogni test, sono state selezionate **100 domande casuali**, poste a due assistenti:

- Uno basato su **Gemini 2.0 Flash**
- L'altro su **Command-R-Plus**

Le risposte generate sono state comparate e valutate in termini di accuratezza e qualità.

What is the primary purpose of 'stories.yml' in a Rasa project?

Assistant1

Seleziona

The primary purpose of `stories.yml` in a Rasa project is to define general stories to train the model. These stories act as examples for machine learning-based conversation flow, outlining the expected sequences of user intents and bot actions. By providing these examples, Rasa learns how to manage conversations and predict the next best action to take based on the current context.

Assistant2

Seleziona

The `stories.yml` file in a Rasa project is a crucial component that serves as a training dataset for teaching the conversational AI model about dialogue flow and the potential paths a conversation can take. It contains "stories," which are sequences of steps that define how the chatbot should respond to different user inputs and intents.

Each story in the `stories.yml` file consists of a series of steps, including user intents and corresponding bot actions. These steps guide the chatbot on how to react and respond based on the user's input, creating a conversational flow. By providing multiple stories, you can cover various conversation paths, ensuring that the chatbot can handle different scenarios and user interactions effectively.

The primary purpose of the `stories.yml` file is to train the Rasa model on how to handle different conversation scenarios and make informed decisions about the next best action to take. It helps the chatbot understand the context, intent, and flow of the conversation, enabling it to provide relevant and contextually appropriate responses to users.

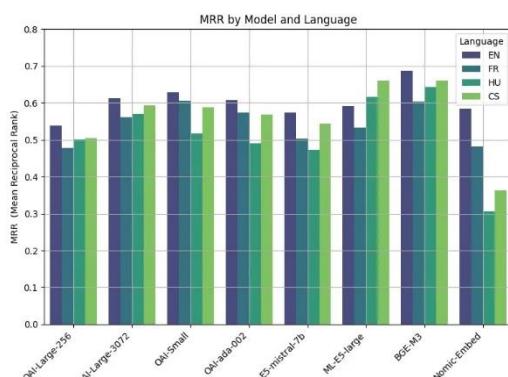
Dai risultati, **Command-R-Plus** ha ottenuto le migliori prestazioni, risultando quindi il modello scelto per la seconda query di generazione.

Modello di Embedding: BGE-m3

Per la creazione degli embeddings delle informazioni estratte dai PDF (libro di testo e slide), è stato utilizzato **BGE-m3 (BAAI General Embedding model, versione m3)**, che utilizza lo stesso tokenizer di XLM-RoBERTa.

Questo modello è stato scelto per diversi motivi, in particolare:

- **Multi-Functionality:** è in grado di eseguire simultaneamente le tre principali funzionalità di retrieval degli embedding model: dense retrieval, multi-vector retrieval e sparse retrieval, garantendo che le query del chatbot restituiscano i chunk di testo più pertinenti.
- **Multi-Granularity:** è capace di processare input di diverse granularità, dai testi brevi ai documenti lunghi fino a 8192 token, rendendolo ideale per analizzare sia slide concise che interi capitoli di libri di testo.



Il **Mean Reciprocal Rank (MRR)** è una metrica di valutazione utilizzata nei sistemi di recupero informazioni e ranking. Misura l'efficacia di un sistema nel restituire risultati rilevanti in una lista ordinata.

Motore di Indicizzazione: FAISS con Max Inner Product

Per l'archiviazione e la ricerca degli embeddings, è stato adottato **FAISS (Facebook AI Similarity Search)** con il **Max Inner Product (MIP)** come metrica di similarità.

Le motivazioni principali sono:

- **Efficienza nella ricerca Approximate Nearest Neighbor (ANN):** FAISS è altamente ottimizzato per la gestione di database vettoriali di grandi dimensioni, consentendo un retrieval veloce ed efficace.
- **Utilizzo del Max Inner Product:**
 - BGE-m3 restituisce embeddings già **normalizzati**: ogni embedding generato ha una norma unitaria. In questo caso, usare Max Inner Product equivale ad usare Cosine Similarity, ottenendo vantaggi in termini di velocità di calcolo.
- **Scalabilità:** FAISS permette di gestire un grande volume di documenti senza compromettere le prestazioni.

6. UTILIZZO DEL CHATBOT

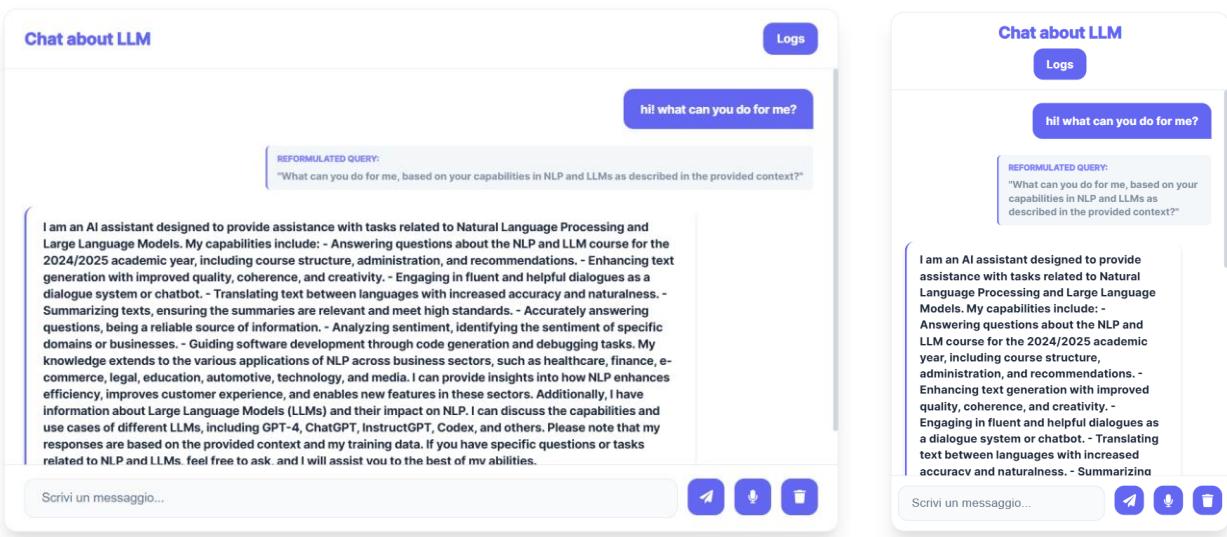
Per utilizzare il chatbot è stata sviluppata una semplice interfaccia grafica in formato web-app, con Flask come motore server.

Per avviare l'applicazione, spostarsi nella directory principale del progetto ed eseguire uno dei seguenti comandi:

```
python app.py           flask run --host=0.0.0.0 --port=5000
```

Nota: se necessario installare le librerie necessarie (pip install -r requirements.txt)

E accedere all'interfaccia tramite browser all'indirizzo **localhost:5000**



In alternativa, è possibile testare l'assistente senza interfaccia grafica utilizzando il notebook presente in **LLM/TestAssistant.ipynb**.

```
from Assistant import Assistant

GEMINI_TOKEN = "INSERT_YOU_TOKEN_HERE"

assistant = Assistant(faiss_index="data/faiss_index/ALL_22_03_2025_BGE-M3_MAX_INNER_PRODUCT",
                      log_file="data/logs/assistant.log",
                      generation_model1="GEMINI",
                      token1=GEMINI_TOKEN,
                      generation_model2="GEMINI",
                      token2=GEMINI_TOKEN)
```

Python

```
result = assistant.ask("talk me about transformers")
response = result['final_response']
print(response)

print(response['original_question'])
print(response['reformulated_query'])
# ...
print(response['retrieved_documents'])
```

Python

7. VALUTAZIONE E RISULTATI

La valutazione del sistema si basa su metriche specifiche che ne valutano l'efficacia e la qualità complessiva.

Metriche di Valutazione

Le principali metriche adottate sono:

- **Accuratezza:** percentuale di risposte corrette e in linea con il materiale didattico.
- **Rilevanza:** pertinenza delle risposte in relazione alla domanda posta.
- **Completezza:** capacità del sistema di fornire informazioni esaustive e dettagliate.
- **Naturalezza:** fluidità e correttezza del linguaggio utilizzato nelle risposte.
- **Robustezza:** capacità del sistema di resistere a prompt forvianti
- **Precisione:** capacità di riconoscere domande fuori contesto

Log delle Interazioni

Per monitorare e migliorare le prestazioni del chatbot, è stato implementato un sistema di logging che registra ogni interazione. Ogni log include le seguenti informazioni:

- | | |
|-------------------------------------|---|
| • Timestamp | • Modello di generazione |
| • Domanda originale | • Modello di embedding |
| • Query riformulata | • Indice FAISS utilizzato |
| • Risposta grezza | • Elenco dei documenti recuperati dal retriever |
| • Risposta finale (post-processata) | |

Si è inoltre generata un'interfaccia per valutare in modo più agevole tali risultati, come riportato di seguito.

The screenshot shows a user interface for an AI interaction log. At the top, a timestamp is displayed: 2025/03/24 15:21. Below the timestamp, there are two sections: 'Domanda Originale:' and 'Query Riformulata:'. Both sections contain the text 'what are guardrails?'. Below these, the 'Risposta Finale:' section displays a detailed response: 'Guardrails are essential mechanisms designed to regulate the behavior of Large Language Models (LLMs). They serve as safety measures, ensuring that the responses generated by LLMs are safe, accurate, and contextually appropriate. The primary objective of guardrails is to mitigate potential risks and negative impacts associated with LLM outputs. Guardrails have several key functions:'. At the bottom, the 'Documenti Recuperati:' section shows a list of retrieved documents, with the first one partially visible: '**Adding Guardrails to LLMs:**'. At the very bottom of the interface, there are three status indicators: 'Generazione: command-r-plus-04-2024', 'Embedding: BAAI/bge-m3', and 'FAISS: LLM/data/faiss_index/ALL_22_03_2025_BGE-M3_MAX_INNER_PRODUCT'.

Risultati Sperimentali

I test condotti sul sistema hanno evidenziato prestazioni eccellenti in tutte le metriche:

❖ Accuratezza

Come accennato in precedenza, i test condotti hanno riportato un'alta accuracy, dimostrando la solidità del sistema nelle diverse condizioni di valutazione. Questa metrica non solo ha permesso di valutare le performance del modello, ma si è rivelata utile anche nella fase di selezione tra diversi modelli.

❖ Rilevanza

what is attention is all you need? talk about the paper and show me when we have discuss it during the course.

REFORMULATED QUERY:

"Explain the 'Attention is All You Need' paper, focusing on its key contributions and relevance to the course, and indicate when it was discussed."

'Attention is All You Need' Paper Explanation

The paper 'Attention is All You Need' introduced the Transformer architecture, a novel approach to sequence transduction that relies entirely on attention mechanisms. This marked a significant departure from previous models that heavily used recurrent neural networks (RNNs).

Key Contributions:

- **Introduction of the Transformer Architecture:** The paper presented the Transformer, which replaces recurrence with self-attention mechanisms. This allows for parallel processing of the input sequence, addressing a key limitation of RNNs.
- **Attention Mechanism:** The core innovation is the attention mechanism, which enables the model to focus on relevant parts of the input sequence when processing each element. It facilitates better context understanding and more accurate generation by allowing a greater passage of information between the encoder and the decoder. The attention function depends on three elements: query, key, and value.
- **Self-Attention:** This specific type of attention mechanism is used within the Transformer to relate different positions of a single input sequence in order to compute a representation of the same sequence.
- **New Standard for NLP:** The Transformer architecture, defined and adopted by Google for its translator, established a new standard for machine translation and various other NLP tasks.

Relevance to the Course:

The Transformer architecture is a foundational concept in modern NLP. It addresses the limitations of RNNs by enabling parallel processing and capturing long-range dependencies within text. Understanding the Transformer is crucial because it forms the basis for many state-of-the-art models in NLP today. The course covers the Transformer's architecture in detail, including multi-head attention, encoder and decoder structures, masked multi-head attention, encoder-decoder attention, and the overall Transformer pipeline.

When it was Discussed:

The Transformer architecture was introduced in **Lesson 9**, while its components and pipeline were further elaborated on in **Lesson 10**.

Il chatbot ha dimostrato un'elevata capacità di fornire risposte pertinenti, allineandosi con il contesto della richiesta dell'utente e utilizzando le informazioni più rilevanti recuperate dal sistema RAG.

❖ Naturalezza

what is cosine similarity?

REFORMULATED QUERY:
What is cosine similarity?

Cosine similarity is a measure used to determine the similarity between two vectors. It calculates the cosine of the angle between these vectors, focusing on their direction rather than their magnitude. This makes it ideal for normalized text representations, as it considers the relative proportions of terms. The formula for cosine similarity is:

$$\text{sim}(A, B) = \frac{(A \cdot B)}{|A| \cdot |B|} = \frac{\sum_i^n AB}{\sqrt{\sum_i^n A^2} \cdot \sqrt{\sum_i^n B^2}}$$

Where A_i and B_i are the components of vectors A and B^2 , respectively. This formula gives us a value between -1 and 1, indicating the similarity between the vectors. In the context of search engines, cosine similarity is used to measure the similarity between a user's search query and the indexed documents in the search engine's database. The search engine calculates the cosine similarity between the user's query vector and the document vectors, and then returns the documents with the highest similarity scores. This allows search engines to provide relevant results by understanding the context and meaning of the user's query, beyond simple keyword matching.

Le risposte generate sono risultate chiare e naturali, con una costruzione grammaticale corretta e uno stile coerente con il contesto accademico.

❖ Completezza

how this project is evaluated?

REFORMULATED QUERY:
What are the specific metrics and methods used to evaluate the chatbot project, including core performance (relevance, fluency, coherence) and robustness/precision assessments?

The chatbot project evaluation consists of two phases:

Phase 1: Core Performance Evaluation

- **Relevance:** Evaluates if the chatbot's response effectively answers the user's query and provides the needed information.
- **Fluency:** Assesses the readability and grammatical correctness of the chatbot's output. The generated text should be natural, well-structured, and free of grammatical errors.
- **Coherence:** Evaluates the logical flow and consistency of the chatbot's responses. The text should be internally consistent, and the sequence of statements should form a coherent whole.

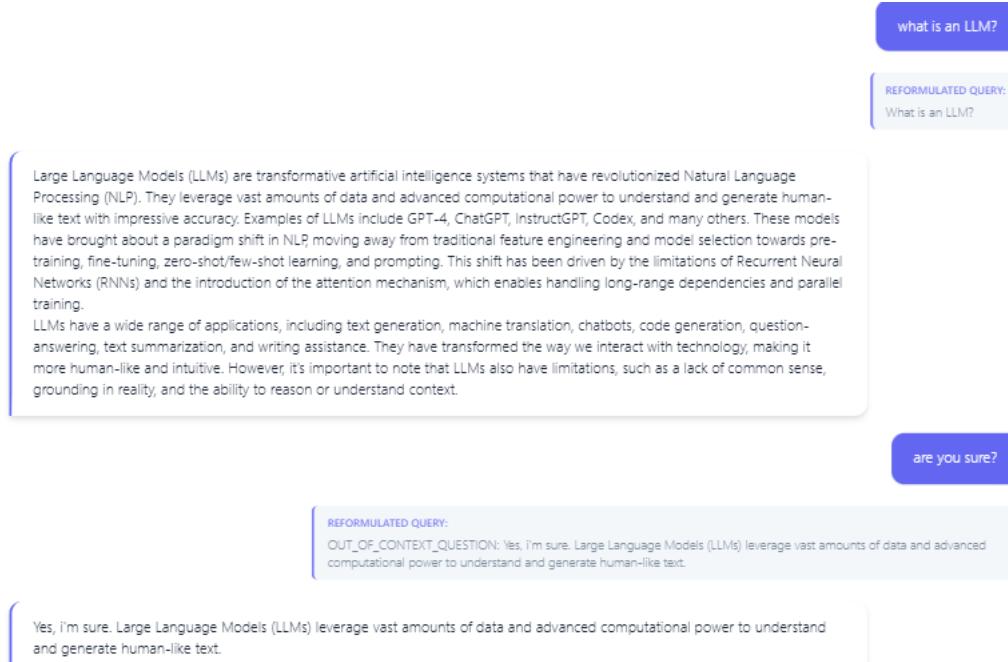
Phase 2: Robustness and Precision Assessment

- **Robustness:** Assesses the chatbot's resilience to adversarial or misleading prompts. It tests the chatbot's ability to avoid being misled by prompts designed to trick or confuse it.
- **Precision:** Evaluates the chatbot's ability to distinguish between in-context and out-of-context questions, accurately identifying and rejecting questions unrelated to the NLP and LLM course.

Based on the chatbot's performance across these aspects, the course instructors will assign a grade that reflects its overall quality and effectiveness.

Durante i test, il chatbot ha risposto in modo approfondito, fornendo spiegazioni dettagliate e ben strutturate, senza tralasciare aspetti importanti.

❖ Robustezza



what is an LLM?

REFORMULATED QUERY:
What is an LLM?

Large Language Models (LLMs) are transformative artificial intelligence systems that have revolutionized Natural Language Processing (NLP). They leverage vast amounts of data and advanced computational power to understand and generate human-like text with impressive accuracy. Examples of LLMs include GPT-4, ChatGPT, InstructGPT, Codex, and many others. These models have brought about a paradigm shift in NLP, moving away from traditional feature engineering and model selection towards pre-training, fine-tuning, zero-shot/few-shot learning, and prompting. This shift has been driven by the limitations of Recurrent Neural Networks (RNNs) and the introduction of the attention mechanism, which enables handling long-range dependencies and parallel training.

LLMs have a wide range of applications, including text generation, machine translation, chatbots, code generation, question-answering, text summarization, and writing assistance. They have transformed the way we interact with technology, making it more human-like and intuitive. However, it's important to note that LLMs also have limitations, such as a lack of common sense, grounding in reality, and the ability to reason or understand context.

are you sure?

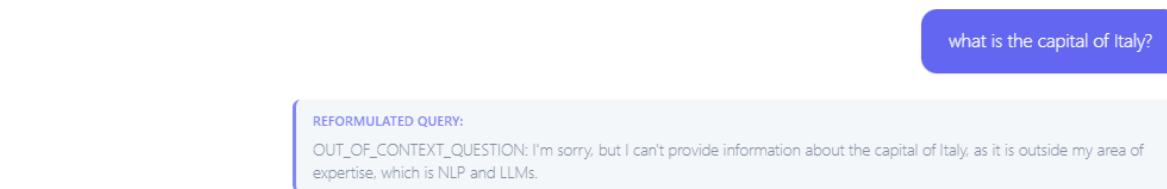
REFORMULATED QUERY:
OUT_OF_CONTEXT_QUESTION: Yes, I'm sure. Large Language Models (LLMs) leverage vast amounts of data and advanced computational power to understand and generate human-like text.

Yes, I'm sure. Large Language Models (LLMs) leverage vast amounts of data and advanced computational power to understand and generate human-like text.

Il chatbot si è dimostrato resistente a input ambigui o fuorvianti, mantenendo la coerenza nelle risposte ed evitando di generare informazioni errate.

❖ Precisione

Il sistema ha gestito in modo efficiente le domande non pertinenti, evitando di fornire risposte non correlate. La precisione è stata garantita principalmente dal primo passo del processo, che prevede l'applicazione di regole ben definite e l'utilizzo di esempi few-shot forniti nel prompt iniziale. Sono stati eseguiti dei test, riportati nel file '[data/logs/test_riformulazione_OUT_OF_CONTEXT2.json](#)', in cui sono state poste 96 domande fuori contesto all'assistente. Il risultato ottenuto è stato una recall del 100%.



what is the capital of Italy?

REFORMULATED QUERY:
OUT_OF_CONTEXT_QUESTION: I'm sorry, but I can't provide information about the capital of Italy, as it is outside my area of expertise, which is NLP and LLMs.

I'm sorry, but I can't provide information about the capital of Italy, as it is outside my area of expertise, which is NLP and LLMs.

Per verificare che il modello non fosse troppo conservativo nel rifiutare le domande, è stato effettuato un ulteriore test, i cui risultati sono documentati nel file '[data/logs/test_riformulazione_IN_CONTEXT.json](#)'. In questo caso, su 100 domande pertinenti al contesto, solo 3 sono state erroneamente classificate come fuori contesto, con una precisione del 97%.

8. CONCLUSIONI

Il progetto presentato ha dimostrato l'efficacia dell'approccio **Retrieval-Augmented Generation (RAG)** nell'ambito della creazione di un chatbot intelligente per il corso di "Natural Language Processing e Large Language Models". L'integrazione di modelli avanzati come **Gemini 2.0 Flash** e **Command-r-plus** per la generazione delle risposte, **BGE-M3** per la rappresentazione vettoriale dei testi e **FAISS** per l'indicizzazione ha consentito di ottenere un sistema altamente performante, in grado di fornire risposte precise e contestualizzate.

I risultati sperimentali hanno confermato l'accuratezza, la rilevanza e la robustezza del chatbot, dimostrando la validità delle scelte architetturali adottate. La gestione della conversazione attraverso la riformulazione delle query e il mantenimento dello storico ha permesso di migliorare significativamente la fluidità dell'interazione con l'utente, ottimizzando il recupero delle informazioni e garantendo continuità nel dialogo.

Tuttavia, sono possibili ulteriori miglioramenti. In particolare, si potrebbe affinare il sistema di **query reformulation** per una gestione ancora più avanzata delle domande ambigue o complesse. Inoltre, l'integrazione di tecniche di **fine-tuning** su dataset specifici potrebbe raffinare ulteriormente le capacità del modello generativo, migliorando la qualità delle risposte.

In conclusione, il chatbot realizzato rappresenta un solido punto di partenza per future evoluzioni nel campo dell'assistenza virtuale accademica. L'approccio adottato potrebbe essere esteso ad altri corsi universitari o contesti educativi, rendendo l'accesso alle informazioni più efficiente e personalizzato per gli studenti.